Original Instructions

PROGRAMMING MANUAL

for

MAZATROL PREVIEW

MANUAL No.: H798PA0015E

Serial No.:

Before using this machine and equipment, fully understand the contents of this manual to ensure proper operation. Should any questions arise, please ask the nearest Mazak service center.

IMPORTANT NOTICE

1. **Be sure to observe the safety precautions described in this manual and the contents of the safety plates on the machine and equipment. Failure may cause serious personal injury or material damage. Please replace any missing safety plates as soon as possible.**
2. **No modifications are to be performed that will affect operation safety. If such modifications are required, please contact the nearest Mazak service center.**
3. **For the purpose of explaining the operation of the machine and equipment, some illustrations may not include safety features such as covers, doors, etc. Before operation, make sure all such items are in place.**
4. **This manual was considered complete and accurate at the time of publication, however, due to our desire to constantly improve the quality and specification of all our products, it is subject to change or modification. If you have any questions, please contact the nearest Mazak service center.**
5. **Always keep this manual near the machinery for immediate use.**
6. **If a new manual is required, please order from the nearest Mazak service center with the manual No. or the machine name, serial No. and manual name.**

Issued by *Manual Publication Section, Yamazaki Mazak Corporation*, Japan

01. 2010

CONTENTS

Page

1 INTRODUCTION 1-1

2 PROGRAMMING 2-1

2-1 Controlled Axes 2-1

2-1-1 Dimension words and controlled axes 2-1

2-1-2 Decimal point input 2-1

2-2 Units of Data Input 2-3

2-2-1 Units of program data input 2-3

2-2-2 Units of setting data input 2-3

2-3 Data Formats 2-3

2-3-1 Program formats 2-3

2-3-2 Optional block skip (BLOCK DELETE) 2-4

2-3-3 Program number, sequence number: O, N 2-5

2-4 Preparatory Function (G-Code) 2-5

2-5 Position Programming 2-7

2-5-1 Dimensional data input method: G90, G91 2-7

2-6 Interpolation Functions 2-8

2-6-1 Positioning (Rapid traverse): G00 2-8

2-6-2 Linear interpolation: G01 2-9

2-6-3 Circular interpolation: G02, G03 2-10

2-6-4 Radius-designated circular interpolation: G02, G03 2-13

2-7 Feed Functions 2-14

2-7-1 Rapid traverse rates 2-14

2-7-2 Rates of cutting feed 2-14
2-7-3 Exact-stop check: G09 2-14
2-7-4 Exact-stop check mode: G61 2-15
2-7-5 Automatic corner deceleration mode: G62 2-15
2-7-6 Continuous cutting mode: G64 2-16
2-7-7 Characteristics of G61, G62 and G64 2-16
2-7-8 Optimum acceleration & deceleration control: G22 2-18
2-8 Dwell Functions 2-19
2-8-1 Dwell command: G04 2-19
2-9 Miscellaneous Functions 2-19
2-9-1 Miscellaneous function (M2-digit) 2-19
2-10 Tool Radius Compensation: G40, G41, G42 2-22
2-10-1 Overview 2-22
2-10-2 Tool radius compensation 2-22
2-10-3 Programmed compensation data input: G10 2-32
2-11 Programming Support Functions 2-33
2-11-1 Subprogram control commands: M98, M99 2-33
2-11-2 Programmed coordinate system rotation: G68, G69 2-35
2-11-3 Scaling: G50, G51 2-36
2-11-4 Programmable mirror image: G50.1, G51.1 2-39
2-11-5 Corner chamfering and rounding 2-40
2-11-6 Polar coordinates input: G16, G15 2-44
2-12 Coordinate System Setting Functions 2-46
2-12-1 Machine coordinate system, workpiece coordinate systems, and local coordinate systems 2-46

2-12-2 Machine zero point and second, third, and fourth reference points 2-46
2-12-3 Coordinate system setting: G92 2-47
2-12-4 Reference point return: G28 2-48
2-12-5 Return to the 2., 3. or 4. reference point: G30 2-48
2-12-6 Workpiece coordinate system selection: G54 to G59 2-49
2-12-7 Local coordinate system setting: G52 2-50
2-12-8 Machine coordinate system selection: G53 2-54
2-13 Laser Control Functions 2-55
2-13-1 Programmed cutting conditions: G82 2-55
2-13-2 Focal point control: G87 2-56
2-13-3 Assisting-gas pressure control: G88 2-57
2-14 Laser Control Programming Example 2-59
2-15 Sample Program for High-Speed Cutting 2-61
3 POWER RAMPING FUNCTION 3-1
3-1 Introduction 3-1
3-1-1 Detailed description 3-1
3-2 Programming 3-2
3-2-1 Program 3-2
3-3 Setting Conditions 3-3
3-4 Parameter 3-3
3-5 Real-time Power Ramping Function 3-4
3-5-1 Sample programs 3-5
3-5-2 Related parameters 3-5

4 MACHINING MODE SELECTION (for the HG and HTX) 4-1

4-1 Overview 4-1

4-2 Programming Format 4-1

4-3 Detailed Description 4-1

4-4 Related Parameters 4-2

5 ELLIPTICAL POSITIONING 5-1

5-1 Overview 5-1

5-2 Programming Format 5-1

5-3 Operation 5-2

5-4 Detailed Description 5-3

5-5 Related Parameters 5-4

5-6 Programming Example 5-5

6 USER MACROS 6-1

6-1 Introduction 6-1

6-2 Macro Call Instructions 6-1

6-3 Variables 6-8

6-3-1 Outline 6-8

6-3-2 Types of variables 6-9

6-3-3 Arithmetic operation commands 6-21

6-4 Control Commands 6-25

7 PROGRAMMABLE FOCUS POSITIONING (SERVO FOCUS) 7-1

7-1 Overview 7-1

7-2 Interpolation Functions 7-1

7-2-1 Positioning (Rapid traverse): G00 7-1

7-2-2 Linear interpolation: G01 7-3

7-2-3 Reference point return: G28 7-3

7-2-4 Machine coordinate system selection: G53 7-4

- NOTE -

1 INTRODUCTION

1. Features

Programming on the NC unit MAZATROL PREVIEW, where conventional G-codes for laser machine tools are utilized, has the following features:

1. Alphanumeric data input from the keyboard.
2. Similar programming to that for conventional laser machine tools with similar G-codes.
3. Control of the Z-axis as required.
4. Setting of cutting conditions (laser output and rate of feed) as required.
5. Special G-codes for programmed coordinate system rotation, scaling, mirror image, etc.
6. Corner chamfering and rounding for any angle.
7. Smallest input capacity (for coordinates, etc.): 0.001 mm or 0.0001 in.

- NOTE -

2 PROGRAMMING

2-1 Controlled Axes

2-1-1 Dimension words and controlled axes

Four machine axes are controlled: the axes X, Y, Z and W*. The Z-axis, however, is mainly “trace” controlled, and the W-axis is used to change the position of the focal point. Simultaneous four-axis control is possible. With the Z-axis being trace-controlled, however, simultaneous two-axis control can be performed (in the XY-plane).

Dimension words comprised of an alphabetical letter and a number are used to specify axis movements in each direction.

* The W-axis is only valid for machines equipped with the Servo Focus function. See Chapter 7 for details on the Servo Focus function.

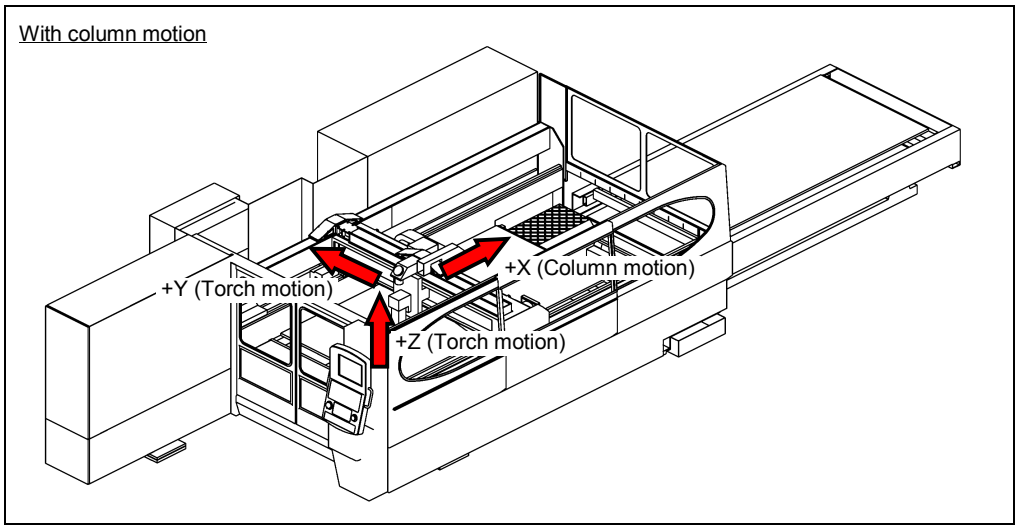


Fig. 2-1

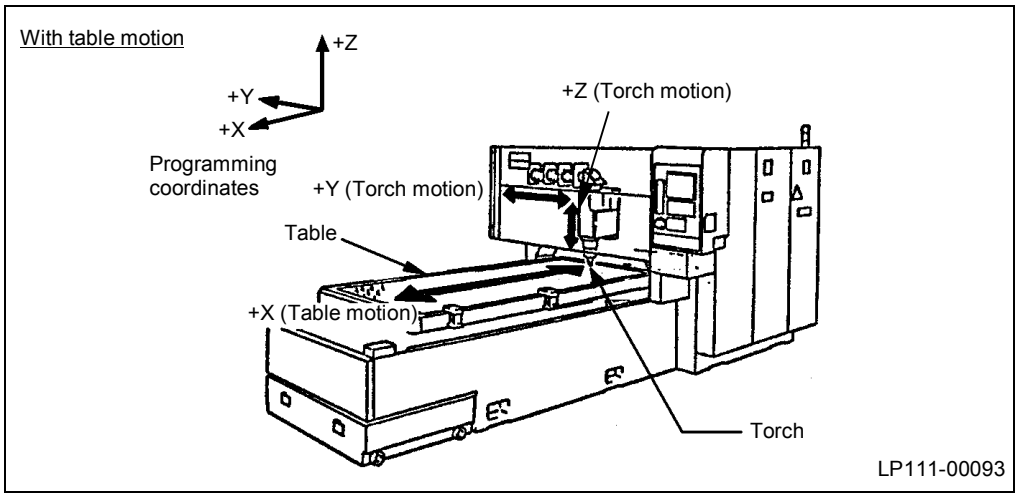


Fig. 2-2

2-1-2 Decimal point input

1. Function and purpose

The decimal point can be used to determine the units digit (mm or inch) of dimensional data.

2. Programming format

OOOO. OOO Metric system
OOO. OOOO Inch system

3. Detailed description

1. The decimal point can be used for the distance, angle, time, speed and scaling factor in machining programs.
2. The unit data “1” without decimal point are processed as follows:

Item		Metric system	Inch system
Axis movement	X1;	0.001 mm	0.0001 inch
Dwell	X1;	0.001 sec	0.001 sec
	P1;	0.001 sec	0.001 sec
Angle	R1;	0.001 deg.	0.001 deg.
Scaling factor	P1;	× 0.001	× 0.001
Rate of feed	F1;	1 mm/min	1 inch/min

3. The decimal point can be used with addresses X, Y, Z, C, I, J, R, F, P and W. Refer to the table below for details.
4. Figures after the decimal point are simply ignored for an address which cannot accept it.

Table 2-1 Decimal point Valid/Invalid for data addresses

Address	Decimal point	Function
C	Valid	Corner chamfering size
D	Invalid	Tool radius compensation number
F	Valid	Rate of feed
G	Invalid	Preparatory function (except G50.1 and G51.1)
I	Valid	Coordinate of arc center
J	Valid	Coordinate of arc center
L	Invalid	Number of subprogram repetitions
M	Invalid	Miscellaneous function
N	Invalid	Sequence number
O	Invalid	Program number
P	Invalid	Subprogram call number
	Invalid	Dwell time
	Valid	Scaling factor
	Invalid	Number of the reference point for return
	Invalid	Laser peak output
	Invalid	Assisting-gas pressure
	Invalid	Focal point offset amount
Q	Invalid	Laser pulse frequency
R	Valid	Arc radius
	Valid	Corner radius
	Valid	Angle of rotation
	Invalid	Laser pulse duty
W*	Valid	W-axis coordinate
X	Valid	X-axis coordinate
	Valid	Dwell time
	Valid	Radius of polar coordinate
Y	Valid	Y-axis coordinate
	Valid	Angle of polar coordinate
Z	Valid	Z-axis coordinate

* The W-axis is only valid for machines equipped with the Servo Focus function. See Chapter 7 for details on the Servo Focus function.

2-2 Units of Data Input

2-2-1 Units of program data input

The units of program data input are the axial movements commanded by MDI and programs. Two systems are available for the description: Metric system and Inch system.

The units digit in data with the decimal point correspond with the unit of notation (mm or inch).

2-2-2 Units of setting data input

The units of setting data input refer to various data commonly used for the controlled axes, such as offsetting data, to be stored in the NC unit.

	Programming unit	Setting unit	Position display sensitivity
Metric system	0.001 mm	0.001 mm	0.001 mm
Inch system	0.0001 in.	0.0001 in.	0.0001 in.

Note 1: The selection between metric and inch systems can be made on the **SETTING** screen.

Note 2: The change in unit system concerns both the programming data input and setting data input. Never fail, therefore, to modify all types of data concerned — programmed dimensions as well as setting parameters (offsetting data, etc.) — according to the change in unit system.

2-3 Data Formats

2-3-1 Program formats

A format predetermined for assigning control information to the NC unit is referred to as a program format. The program format used for our NC unit is word address format.

1. Words and addresses

A word is a set of characters arranged as shown below, and information is processed in words.

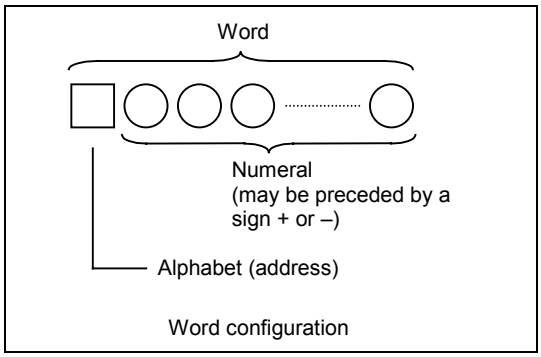


Fig. 2-3

The alphabetic character leading a word is referred to as an address, which defines the meaning of the succeeding numeric information. (Refer to Table 2-2 for the details of the word address format.)

2. Blocks

A block, unit of instruction, contains a number of words which constitute information necessary for the NC machine to perform an operation. The end of each block must be indicated by an EOB (End Of Block) code.

3. Programs

A number of blocks form one program.

Table 2-2 Details of the word address format

Item and address		Metric system	Inch system
Program No.:	O	1 - 9999	
Sequence No.:	N	0 - 9999	
Preparatory function:	G	0 - 99	
Basic axes	X, Y	±99999.999 mm	±9999.9999 in.
	Z	±99999.999 mm	±9999.9999 in.
	W*	±99999.999 mm	±9999.9999 in.
Auxiliary axes	I, J, C	±99999.999 mm	±9999.9999 in.
	R	±99999.999 mm	±9999.9999 in.
Dwell	X	0 - 99999.999 sec	0 - 99999.9999 sec
	P	0 - 99999999 msec	0 - 99999999 msec
Rate of feed	F	1 - 99999.999 mm/min	1 - 9999.9999 in./min
Miscellaneous function:	M	0 - 99	
Subprogram No.:	P	1 - 9999	
Number of repetitions:	L	0 - 9999	
Tool offset No.:	D	0 - 32	
Laser processing conditions	Output: P	1 - 9999 W (Depending on the oscillator specification)	
	Frequency: Q	1 - 9999 Hz (Depending on the oscillator specification)	
	Pulse duty: R	1 - 100%	
	Gas pressure: P	1 - 99 (× 0.1 kgf/cm^2) (Depending on the machine specification)	

* The W-axis is only valid for machines equipped with the Servo Focus function. See Chapter 7 for details on the Servo Focus function.

2-3-2 Optional block skip (BLOCK DELETE)

1. Function and purpose

Optional block skip is a function that selectively ignores that specific block within a machining program which begins with the slash code "/".

Any block beginning with "/" will be ignored if the **BLOCK DELETE** menu function is set to ON, or will be executed if the menu function is set to OFF.
For example, if all blocks are to be executed for a type of parts but specific blocks are not to be executed for another type, then different parts can be machined using one and the same program that contains the "/" code at the beginning of the specific blocks.

2. Operating notes

1. The slash (/) must be placed at the beginning of a block to effectively use the menu function.

   Example: `/N20 G01 X25. Y25.;`

2. Irrespective of whether BLOCK DELETE is on, the vertical parity check is surely performed on slashed blocks as well.
3. BLOCK DELETE processing occurs during prereading. Blocks that have already been stored into the pre-read buffer cannot be skipped.
4. The menu function does not affect program loading and saving operations. Blocks with a leading slash are surely loaded and saved even when the BLOCK DELETE in on.

2-3-3 Program number, sequence number: O, N

Program numbers and sequence numbers are used to monitor the execution status of a machining program or to call a machining program or a specific process within a machining program.

Program numbers identify programs on a workpiece-by-workpiece or subprogram-by-sub-program basis. A program number must be set using the letter O (address) and a numeric of a maximum of four digits.
Sequence numbers are assigned to command blocks as required. A sequence number must be set using the letter N (address) and a numeric of a maximum of four digits.

2-4 Preparatory Function (G-Code)

Preparatory functions, or G-codes, that are classified into groups according to their specific functions, can be distinguished further between modal and non-modal functions.

1. Non-modal G-codes
   Only act on the block in which they are given.
2. Modal G-codes
   Remain valid until they are cancelled or replaced by another G-code of the same group.

Note 1: The symbol ▲ in the following table refers to automatic modal validation by switching-on or resetting.

Note 2: Multiple G-codes of different modal groups can be given in the same block.

Note 3: Non-modal codes form the G-code group 0.

Table 2-3 List of G-codes

G-code	Group	Function	Programming format
▲G00	1	Positioning (rapid traverse)	`G00 X_Y_Z_;`
G01		Linear interpolation (cutting feed)	`G01 X_Y_Z_F_;`
G02		Circular interpolation [CW]	`G02 (G03) X_Y_I_J_F_;`
G03		Circular interpolation [CCW]	`G02 (G03) X_Y_R_F_;`
G04	0	Dwell	`G04 X_(P_);`
G09		Exact-stop check	`G09;`
G10		Programmed compensation data input	`G10 D_R_;`
G15	17	Polar coordinates input	`G15;`
G16		Polar coordinates input OFF	`G16;`
G22	0	Optimum acceleration & deceleration	`G22 J_;`
G28		Reference point return	`G28 X_Y_Z_;`
G30		Return to the 2., 3. or 4. reference point	`G30 P_X_Y_Z_;`
▲G40	7	Tool radius compensation cancel	`G40 X_Y_;`
G41		Tool radius compensation [to the left]	`G41 (G42) X_Y_D_;`
G42		Tool radius compensation [to the right]	
▲G50	11	Scaling cancel	`G50;`
G51		Scaling	`G51 X_Y_P_;`
▲G50.1	18	Programmable mirror image cancel	`G50.1 X_Y_;`
G51.1		Programmable mirror image	`G51.1 X_Y_;`
G52	0	Local coordinate system setting	`G52 X_Y_Z_;`
G53		Machine coordinate system selection	`G53 X_Y_Z_;`

G-code	Group	Function	Programming format
G54	14	Selection of workpiece coordinate system 1	(G90) G54 X_Y_Z_;
G55		Selection of workpiece coordinate system 2	
G56		Selection of workpiece coordinate system 3	
G57		Selection of workpiece coordinate system 4	
G58		Selection of workpiece coordinate system 5	
G59		Selection of workpiece coordinate system 6	
G61	15	Exact-stop check mode	G61;
G62		Automatic corner deceleration mode	G62;
▲G64		Continuous cutting mode	G64;
G65	0	User-macro simple call	* See Chapter 6 for details.
G66	12	User-macro modal call	
▲G67		User-macro modal call cancel	
G68	16	Programmed coordinate system rotation	G68 X_Y_R_;
▲G69		Programmed coordinate system rotation OFF	G69;
G81	13	Power ramping ON	G81 P_Q_R_F_;
▲G83		Power ramping OFF	G83;
G82	0	Programmed cutting conditions	G82 P_Q_R_F_;
G87		Programmed gap offset	G87 P_;
G88		Programmed assisting-gas pressure	G88 P_M_;
▲G90	3	Absolute data input	G90 (G91) X_Y_Z_;
G91		Incremental data input	
G92	0	Coordinate system setting	G92 X_Y_Z_;
G108		Machining mode selection	* See Chapter 4 for details.
G124	33	Piercing detection ON	* See the Operating Manual for INTELLIGENT LOOP SYSTEM for details.
▲G125		Piercing detection OFF	
G135	34	Burning/Plasma detection ON	
▲G136		Burning/Plasma detection OFF	
▲G150	30	Elliptical positioning OFF	* See Chapter 5 for details.
G153		Elliptical positioning ON	
G181	10	Real-time power ramping ON	G181 R_;
▲G183		Real-time power ramping OFF	G183;

2-5 Position Programming

2-5-1 Dimensional data input method: G90, G91

1. Function and purpose

Setting of G90 or G91 allows succeeding dimensional data to be processed as absolute data or incremental data.

2. Detailed description

1. In the absolute data mode, axis movement will be performed to the program-designated position within the workpiece coordinate system, irrespective of the current position.

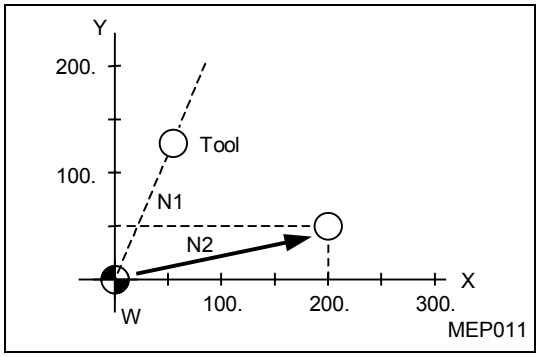


Fig. 2-4

```
N1 G90G00X0 Y0;
```

In the incremental data mode, axis movement will be performed through the program-designated distance as relative data with respect to the current position.

```
N2 G91G01X200. Y50. F100;
```

```
N2 G90G01X200. Y50. F100;
```

Commands for a movement from the origin of the workpiece coordinate system are given with the same values, irrespective of whether the absolute data mode or the incremental data mode is used.

2. The last G90 or G91 command works as a modal one for the following blocks.

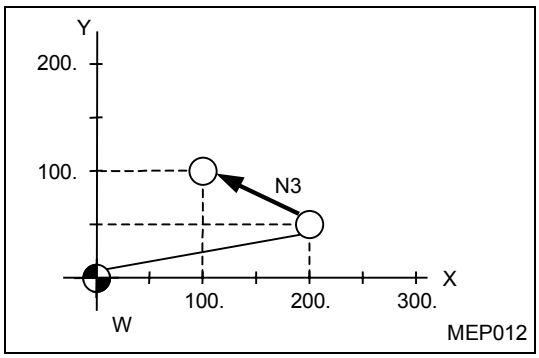


Fig. 2-5

(G90) `N3 X100. Y100.;`

This block will perform a movement to the position of X = 100 and Y = 100 in the workpiece coordinate system.

(G91) `N3 X-100. Y50.;`

This block will perform a movement of –100 on the X-axis and +50 on the Y-axis, and thus result in a movement to the position of X = 100 and Y = 100.

3. The absolute data input is automatically selected by switching-on or resetting.
4. Even in the MDI (Manual Data Input) mode, G90 and G91 will also be handled as modal commands.

2-6 Interpolation Functions

2-6-1 Positioning (Rapid traverse): G00

1. Function and purpose

Positioning command G00 performs a linear positioning (generally in a polygonal line) to the ending point specified by a coordinate word.

2. Programming format

```
G00  Xx Yy Zz;
```

where x, y and z each denote a coordinate. The absolute or the incremental data input is used according to the status of G90/G91 existing at the particular time.

3. Detailed description

1. Once this command has been given, the G00 mode will be retained until any other G-code command that overrides this mode, that is, G01, G02, or G03 of group 01 is given. Thus, it is merely required to input coordinate words for G00 movements in the succeeding blocks.
2. In the G00 mode, acceleration/deceleration always takes place at the starting/ending point of a block and, after an in-position status check, the program proceeds to the next block.

4. Sample programs

```
G91  G00 X–270. Y300. Z150.;
```

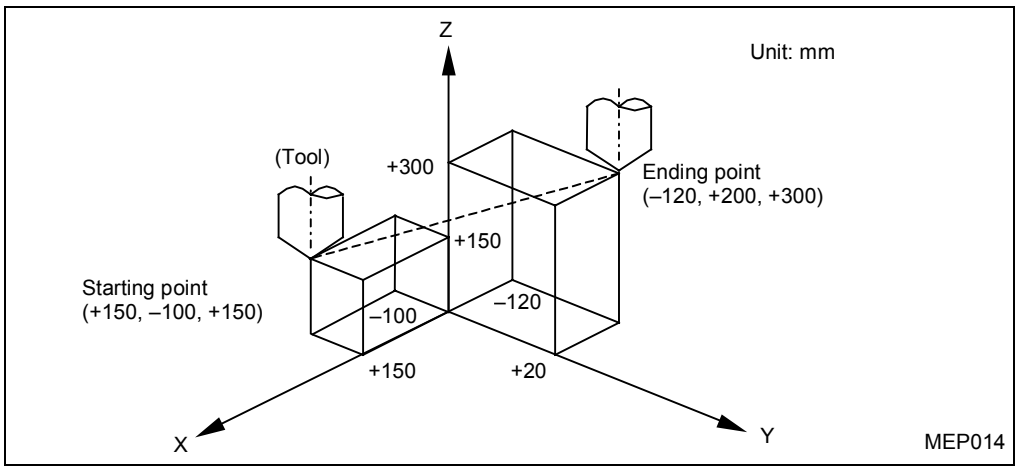


Fig. 2-6

Note 1: When “Non-interpolation for G0” is not selected on the **USER PARAMETER** screen, the “tool” will take the shortest path connecting the starting and ending points. The positioning speed will be distributed automatically to each axis to obtain the shortest positioning time on condition that the rapid traverse rate of any axis is not exceeded.

Note 2: Normally (with the above parameter item being selected) the “tool” will be moved independently at the rapid traverse rate of each axis. If a rapid traverse rate of 9600 mm/min is preset for both X- and Y-axes, then the command

```
G91 G00 X–300. Y200.;
```

will be executed as illustrated below.

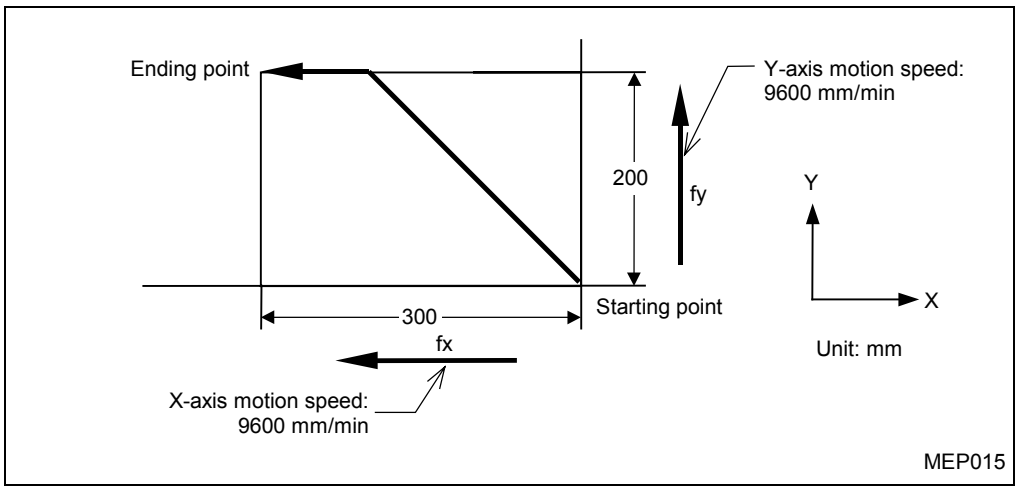


Fig. 2-7

Note 3: Refer to the specifications of the particular machine for the actual rate of rapid traverse.

2-6-2 Linear interpolation: G01

1. Function and purpose

This command moves (interpolates) a tool from the current position to the ending point specified by a coordinate word, at the rate of feed specified with address F. The specified speed acts here as the linear velocity relative to the direction of movement of the tool center.

2. Programming format

```
G01  Xx Yy Zz Ff;
```

where x, y and z each denote a coordinate. The absolute or the incremental data input is used according to the status of G90/G91 existing at the particular time.

3. Detailed description

Once this command has been given, the G01 mode will be retained until any other G-code command that overrides this mode, that is, either G00, G02, or G03 of command group 01 is given. Thus, it is merely required to input coordinate words for linear interpolations in the succeeding blocks unless the rate of feed must be changed.
A programming error will result if no F-code command has been given to the first G01 command.

4. Sample program

The following shows a program for moving the tool at a rate of feed of 300 mm/min on the route of $P_1 \rightarrow P_2 \rightarrow P_3 \rightarrow P_4 \rightarrow P_1$ (route $P_0 \rightarrow P_1$ is of rapid positioning):

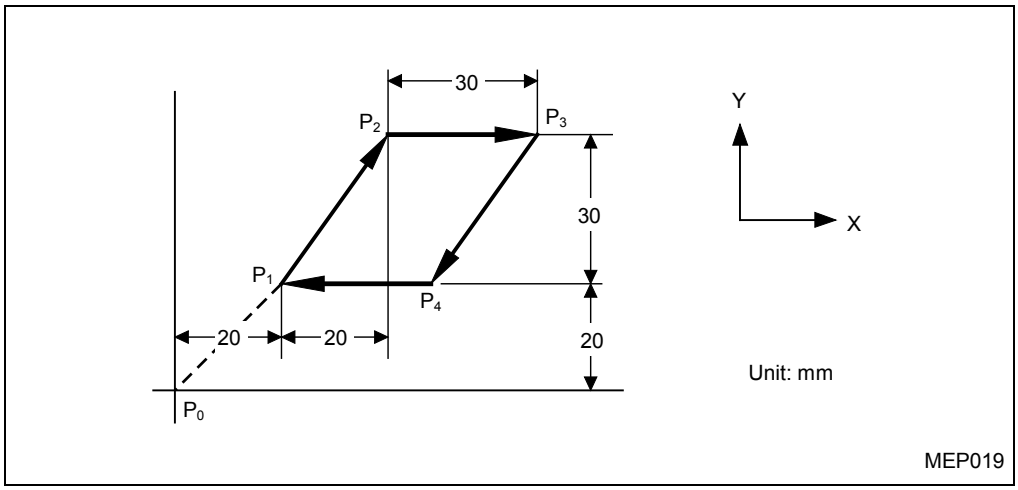


Fig. 2-8

```
G91   G00  X20.   Y20.;          P0 → P1
      G01  X20.   Y30. F300;     P1 → P2
           X30.;                 P2 → P3
           X-20.  Y-30.;         P3 → P4
           X-30.;                P4 → P1
```

2-6-3 Circular interpolation: G02, G03

1. Function and purpose

Commands G02 and G03 feed the tool along an arc.

2. Programming format

```
G02(G03) Xx Yy Ii Jj Ff;
```

where G02/G03 : Circular movement direction CW/CCW
x, y : Coordinates of ending point
i, j : Coordinates of arc center
f : Rate of feed

Use addresses X and Y to specify the coordinates of the ending point of arc, and addresses I and J for the coordinates of arc center.
Both absolute and incremental data input methods are available for setting the coordinates of the ending point of arc. For the coordinates of the arc center, however, incremental data relative to the starting point must always be set.

3. Detailed description

1. Once the G02 (or G03) command has been given, this command mode will be retained until any other G-code command used to override the G02 (or G03) command mode, that is, G00 or G01 of group 01 is given.

2. The direction of circular movement is determined by G02/G03.

   G02: CW (Clockwise)
   G03: CCW (Counterclockwise)

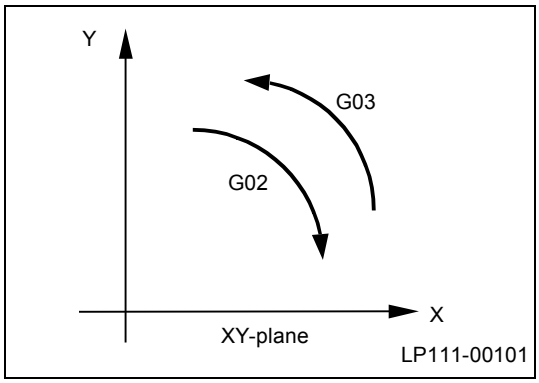


Fig. 2-9

3. Interpolation of an arc that spans multiple quadrants can be defined with one block.

4. Sample programs

Example 1: `G02 J50.F500;` ..... Complete-circle command

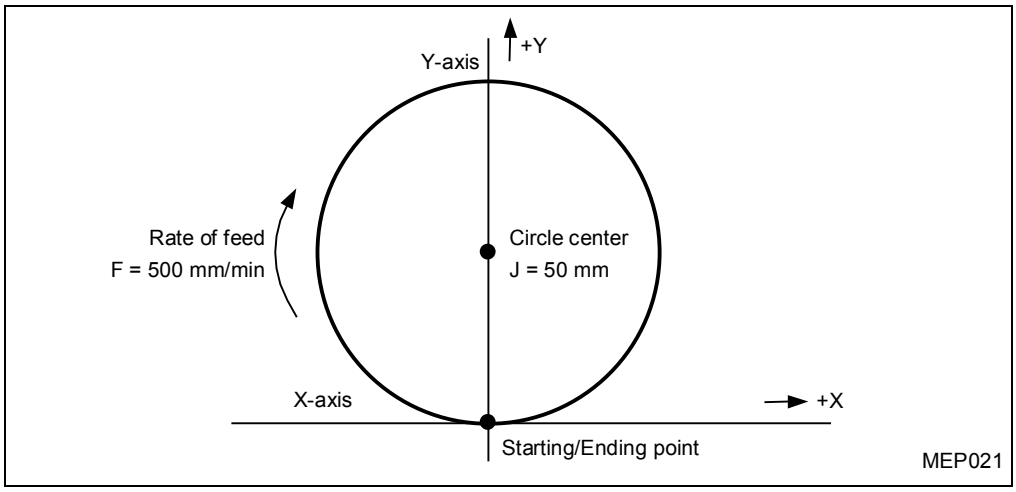


Fig. 2-10

Example 2: `G91 G02X50.Y50.J50.F500;` ..... Three-quarter circle command

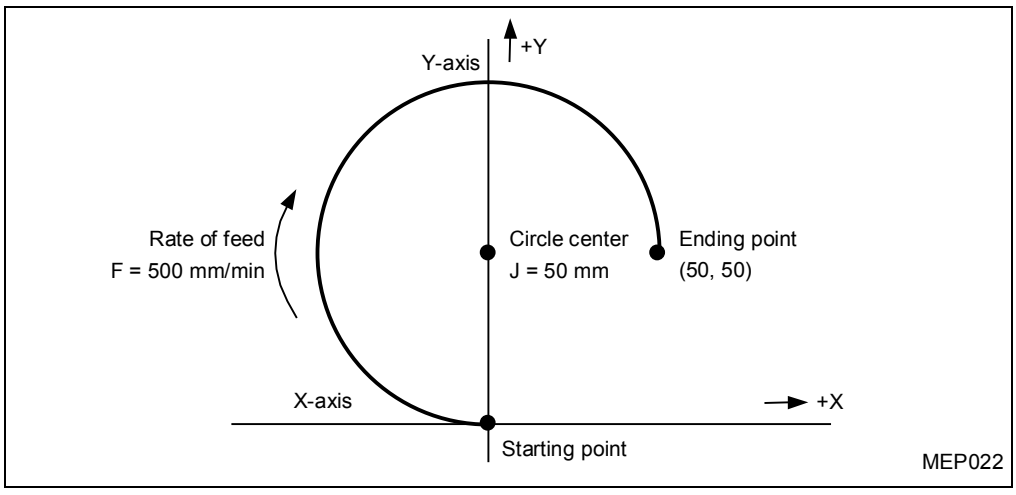


Fig. 2-11

5. Remarks

- If the coordinates of the ending point are all omitted or set to the same position as the starting point, designating the center using addresses I and J will result in an arc of 360 degrees (complete circle).
- Either 1 or 2 below will result if the starting-point radius and the ending-point radius are not the same.

1. If error ⊿R is larger than the parameter data, a program error will occur at the starting point of the arc.

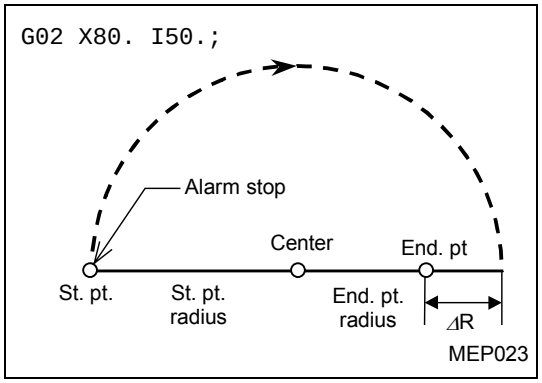


Fig. 2-12

2. If error ΔR is equal to or smaller than the parameter data, interpolation path will be drawn as follows to the programmed ending point of the arc.

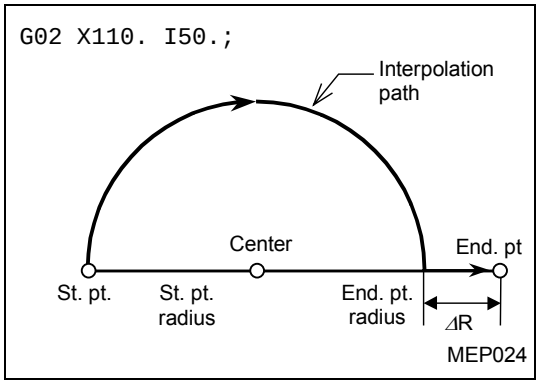


Fig. 2-13

Error ΔR can be set in a parameter (standard setting is 50 micrometers). The examples shown above assume that excessively large parameter data is given to facilitate your understanding.

2-6-4 Radius-designated circular interpolation: G02, G03

1. Function and purpos

Circular interpolation can be performed by designating directly the arc radius R as well as specifying conventional arc center coordinates (I, J).

2. Programming format

```
G02(G03) Xx Yy Rr Ff;
```

where x : X-axis coordinate of the ending point
y : Y-axis coordinate of the ending point
r : Radius of the arc
f : Rate of feed

3. Detailed description

The arc center is present on the mid-perpendicular of the segment which connects the starting point and the ending point. The intersecting point of the mid-perpendicular and the circle of the designated radius **r** around the starting point gives the center coordinates of the designated arc.
A semi-circle or smaller will be generated if R is a positive value.
An arc larger than a semi-circle will be generated if R is a negative value.

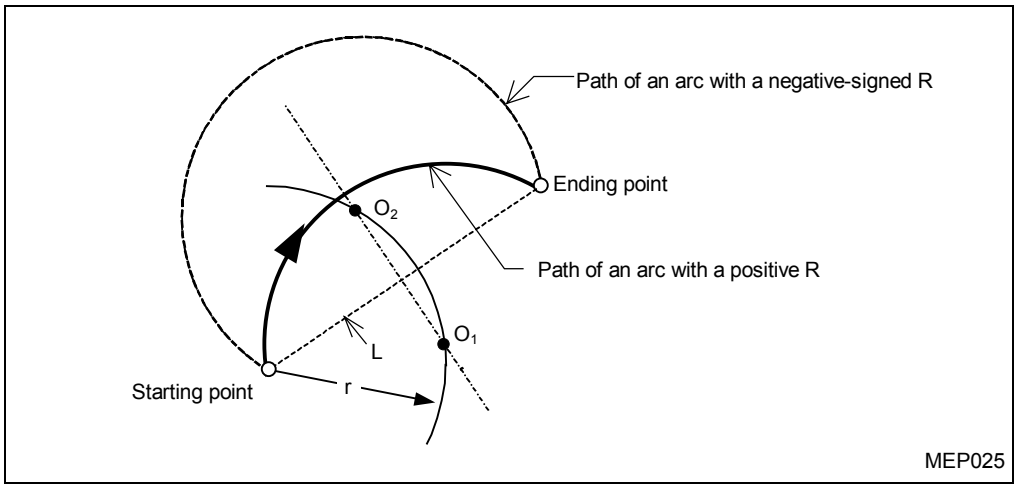


Fig. 2-14

To use the radius-designated circular interpolation commands, the following requirement must be met:

$$\frac{L}{2 \bullet r} \leq 1$$

where L denotes the length of the line from the starting point to the ending point.
If radius data and arc center data (I, J) are both set in the same block, then circular interpolation by radius designation will have priority in general.
For complete-circle interpolation (the ending point = the starting point), however, use center-designation method with addresses I and J, since the radius-specification command in this case will immediately be completed without any machine operation.
Select a plane for the radius-designated circular interpolation in the same manner as for center-specification method.

2-7 Feed Functions

2-7-1 Rapid traverse rates

A separate rapid traverse rate can be set for each axis. The range of rapid traverse rates which can be set is from 1 to 50000 mm/min (0.04 to 1968.50 in./min). The maximum speed, however, is limited according to the particular machine specifications.
Rapid traverse rates apply for commands G00, G28 and G30.

According to the related parameter, the rapid positioning path is either of linear-interpolation type or non-interpolation type (independent rapid motion along each axis). There is no difference, however, in positioning time between the two types.

2-7-2 Rates of cutting feed

A rate of cutting feed must be designated using address F and an eight-digit number (F8-digit direct designation).
The F8 digits can consist of five integral digits and three decimal digits, with the decimal point. The least significant digit in an F-command without decimal point must correspond with the units digit for the assignment of the decimal point.

Rates of cutting feed apply for commands G01, G02 and G03.

Example: `G01 X100. Y100. F200;` for 200 mm/min.
(`F200.` or `F200.000` specifies the same rate of feed.)

Setting range for rate of cutting feed

Command mode	F command range	Range of rate of cutting feed
Metric system	0.001 to 99999.999	0.001 to 99999.999 mm/min
Inch system	0.0001 to 9999.9999	0.0001 to 9999.9999 inches/min

Note 1: A program error will result if no F-code command has been set for the first cutting command (G01, G02, G03) after switching-on.

Note 2: The maximum available rate of feed depends upon the specification of the machine. An F-command in excess of the specification only sets the highest possible speed.

2-7-3 Exact-stop check: G09

1. Function and purpose

Only after the in-position status has been checked following axis motion stop, may you want to start the next block command in order to reduce possible machine shocks due to abrupt changes in rate of feed and to minimize any rounding of workpieces during corner cutting. An exact-stop check function is provided for these purposes.

2. Programming format

```
G09;
```

Exact-stop check command G09 is valid only for the cutting command code (G01, G02 or G03) given in the same block.

3. Sample program

```
N001 G09 G01 X100.F150; ..... The next block is executed after an in-position status check
                             following machine stop.
N002         Y100.;
```

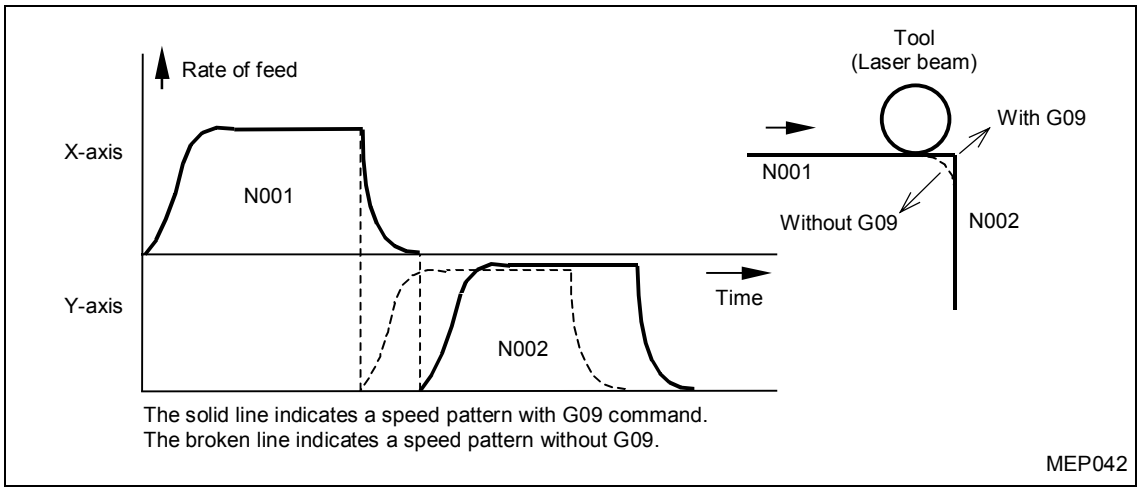


Fig. 2-15 Effect of exact-stop check

2-7-4 Exact-stop check mode: G61

1. Function and purpose

While the exact-stop check command G09 performs the in-position status check merely on the respective block, code G61 functions as a modal command. All blocks of cutting command (under G01, G02 or G03) that succeed G61 are subject to the in-position check.
The modal function of G61 can be cancelled by a G-function in the same group, that is, G62 or G64.

2. Programming format

```
G61;
```

2-7-5 Automatic corner deceleration mode: G62

1. Function and purpose

In the mode of G62 a deceleration to the speed specified in the following parameter is performed at the end of each block.

User parameter **N074**: G62 corner speed (Unit: mm/min)

The modal function of G62 can be cancelled by a G-function in the same group, that is, G61 or G64.

2. Programming format

```
G62;
```

2-7-6 Continuous cutting mode: G64

1. Function and purpose

In the mode of G64 the cutting feed is performed continuously without deceleration at the end of the block.

The modal function of G64 can be cancelled by a G-function in the same group, that is, G61 or G62.

2. Programming format

```
G64;
```

2-7-7 Characteristics of G61, G62 and G64

1. Exact-stop check mode: G61

- Recommended speed: F0 - F3000
- Effects on production quality:

  High accuracy even at high speed.

  As a deceleration to complete stop is required at the end of every motion block, evil thermal effects (weld penetration, burning, dross) are likely to be produced by the laser beam.
- Recommended work: High-accuracy sheet metal working

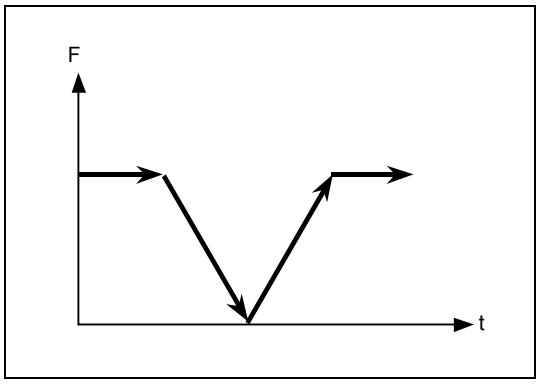


Fig. 2-16

2. Automatic corner deceleration mode: G62

- Recommended speed: F0 - F5000
- Effects on production quality:

  Slight deterioration in accuracy even at high speed.

  As a deceleration to some extent only is required at the end of every motion block, evil thermal effects (weld penetration, burning, dross) by the laser beam are relatively slight.
- Recommended work: High-speed sheet metal working

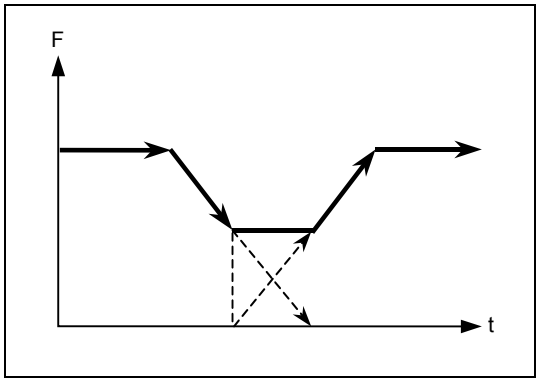


Fig. 2-17

3. Continuous cutting mode: G64

- Recommended speed: F0 - F1500
- Effects on production quality:

  As no deceleration has to occur at the end of every motion block, evil thermal effects (weld penetration, burning, dross) by the laser beam can be reduced to a minimum.

  Faulty shape and vibration may be produced at a higher speed (than 1500 mm/min).
- Recommended work: Thick plate cutting (t6 or more)

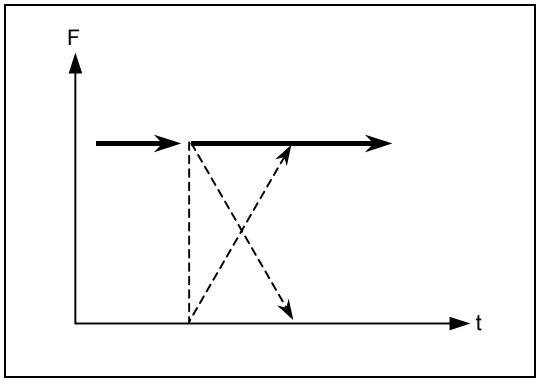


Fig. 2-18

2-7-8 Optimum acceleration & deceleration control: G22

1. Function and purpose

According to the specification of a plate thickness in a block of G22, the acceleration best suitable for the plate thickness is automatically selected: higher acceleration for thinner plates to shorten the machining time, and lower acceleration for thicker and heavier plates to reduce mechanical shock.

2. Programming format

```
G22 J_;
```

where J = plate thickness (in millimeters or inches, decimal point available).

3. Precautions

- Take special care not to give a G22 command for thin plates in machining a thicker plate. Otherwise, the worksheet may be displaced and fall off the clamp, resulting in a serious damage to the machine.
- Give a G22 command at the beginning of the program (prior to the first axis motion). In default of G22 command, the acceleration and deceleration for thick plates is valid for operational safety.
- The manual operation of the machine is executed with the acceleration and deceleration for thick plates.
- When micro-jointless parts are to be machined from a sheet metal plate, it is recommended to use the acceleration type for thick plates, instead of utilizing a G22 command, in order to minimize the displacement of cutoff parts by reducing the mechanical shock during axis movement.

2-8 Dwell Functions

2-8-1 Dwell command: G04

1. Function and purpose

The start of execution of the next block can be delayed using a G04 command.

2. Programming format

```
G04 Xx;
```
or
```
G04 Pp;
```

Data can be set in 0.001 seconds.
For address P, the decimal point is not available.

3. Detailed description

1. The setting range for dwell time is as follows:

Unit of data setting	Range for address X	Range for address P
Metric system	0.001 to 99999.999 (sec)	1 to 99999999 (× 0.001 sec)
Inch system	0.001 to 99999.999 (sec)	1 to 99999999 (× 0.001 sec)

2. The count for the dwell command which is preceded by a block with cutting-feed command is not started until the movement of the preceding block has been brought to a complete stop.

4. Sample programs

```
G04 X500;       Dwell time = 0.5 sec
G04 X5000;      Dwell time = 5.0 sec
G04 X5.;        Dwell time = 5.0 sec
G04 P5000;      Dwell time = 5.0 sec
```

2-9 Miscellaneous Functions

2-9-1 Miscellaneous function (M2-digit)

Miscellaneous functions, which are also referred to as M-code functions, are used to give tracing ON/OFF, discharge ON/OFF commands, and other auxiliary commands for the NC machine.

For the NC unit, these functions must be selected using M2-digit data, and only one M-code can be included in one block.

Example: `G00 Xx Mm;`

If two or more sets of M2-digit data are set, only the last one will become valid.

For M-codes M00, M01, M02 and M30, the next block of data is not read into the input buffer since pre-reading is disabled automatically.

The M-codes can be included in any block that contains other command codes. If, however, the M-code is included in a block that contains movement commands, then the execution priority will be either

- the M-code is executed after completion of movement, or

- the M-code is executed together with movement.

It depends on the machine specifications which is applied.

The following describes six types of special M-code functions:

1. Programmed stop: M00

   The automatic operation will be stopped after execution of the block that contains an M00 command. The modal information remains active, as is the case with a single-block operation mode, and the machine operation can be restarted by pressing the CYCLE START button on the operating panel.

2. Optional stop: M01

   On condition that the **OPTIONAL STOP** menu function is selected, the automatic operation will be stopped after execution of the block that contains an M01 command, as is the case with M00.

   Example:

```
    ⋮
N10 G00 X100.;
N11 M01;
N12 G01 X200.F600;
    ⋮
```

   [OPTIONAL STOP menu function status and operation]
   If the menu function is on, operation stops at N11.
   If the menu function is off, operation does not stop at N11 and N12 is executed.

3. Program End: M02 or M30

   Usually, the program end command is given in the final block of machining program. The automatic operation is terminated and the NC unit is automatically reset.
   Automatic resetting by this command cancels both modal commands and offsetting data, but the position counters are not cleared to zero.

4. Subprogram call/end: M98/M99

   Use M98 or M99 to branch the control into a subprogram or to recall it back to the calling program. Refer to subsection 2-11-1 for details.

The M-codes mentioned above refer to standard functions of the NC unit.
The following table enumerates all the M-codes to be used for machines equipped with the MAZATROL PREVIEW.

Table 2-4 M-code list B: Standard ×: Not available OP: Optional

M-code	Function	HG
M00	Programmed stop	B
M01	Optional stop	B
M02	Program end	B
M09	Assisting gas OFF	B
M20	Laser shutter OPEN	B
M21	Laser shutter CLOSE	B
M22	Laser discharge ON	B
M23	Laser discharge OFF	B
M30	Program end (Reset and rewind)	B
M31	Tracing control OFF	B
M32	Tracing control ON	B
M33	Tracing control only in approaching	B
(M50)	Oxygen as assisting gas at high pressure (*3)	B
(M51)	Oxygen as assisting gas at low pressure (*3)	B
(M52)	Air as assisting gas at high pressure (*3)	B
(M53)	Air as assisting gas at low pressure (*3)	B
M54	3rd/4th gas as assisting gas at high pressure (*1, *2)	B
(M55)	3rd/4th gas as assisting gas at low pressure (*1, *3)	B
M56	Selection of oxygen as assisting gas (*4)	B
M57	Selection of air as assisting gas (*4)	B
M58	Selection of 3rd/4th gas as assisting gas (*1, *4)	B
M60	Side air blow ON	OP
M61	Side air blow OFF	OP
M63	Cancellation of the 4th gas as assisting gas	OP
M64	Selection of the 4th gas as assisting gas	OP
M65	Oil mist OFF	OP
M66	Oil mist ON	OP
M70	Pallet change	B
M77	Spare M-code	OP
M78	Spare M-code	OP
M79	Spare M-code	OP
M81	Loading cycle	OP
M82	Unloading cycle	OP
M83	Unloading & loading cycle	OP
M84	Loader preparation	OP
M98	Subprogram call	B
M99	Subprogram end	B

*1: The fourth gas is only available on a machine equipped with the corresponding option.
*2: With a high-pressure electropneumatic regulator: Pressure command G88P###
With a manual regulator: Pressure command Manual regulator
In other cases: Pressure command Parameter
*3: Pressure command: Parameter
*4: Pressure command: G88P###

2-10 Tool Radius Compensation: G40, G41, G42

2-10-1 Overview

1. Function and purpose

Offsetting in any vectorial direction (in the XY-plane) can be done according to the tool radius preselected using the G-code G41 or G42 and a D-code. This function is referred to as tool radius compensation.

2. Programming format

Programming	Function
`G40`	To cancal the tool radius compensation
`G41 Xx Yy Dd;`	To compensate for tool radius (to the left)
`G42 Xx Yy Dd;`	To compensate for tool radius (to the right)

3. Detailed description

The maximum available number of compensation data numbers is 64: D01 to D64.

Amounts of compensation are to be stored on the **BEAM OFFSET** screen.
The compensation is carried out in the XY-plane.

2-10-2 Tool radius compensation

1. Cancellation of tool radius compensation

Tool radius compensation is cancelled in the following cases:

- After power has been turned on,
- After the reset key on the NC operating panel has been pressed,
- After M02 or M30 has been executed, or
- After G40 (compensation cancellation command) has been executed.

In the compensation cancellation mode, the offset vector becomes zero and the tool center path agrees with the programmed path.
Programs containing tool radius compensation commands must be terminated during the compensation cancellation mode.

2. Startup of tool radius compensation

Tool radius compensation will begin when all the following three conditions are met:

- Command G41 or G42 has been executed.
- The compensation data number for tool radius compensation is larger than zero, but equal to or smaller than the maximum available compensation data number.
- The command used with the compensation command is a move command other than those for circular interpolation.

Compensation will be performed only when reading of several blocks in succession is completed, irrespective of whether continuous operaiton mode or single-block operation mode is used.

In the descriptive diagrams below, “s” signifies the stop point of single-block operation.

3. Starting operation for tool radius compensation

A. For the corner interior

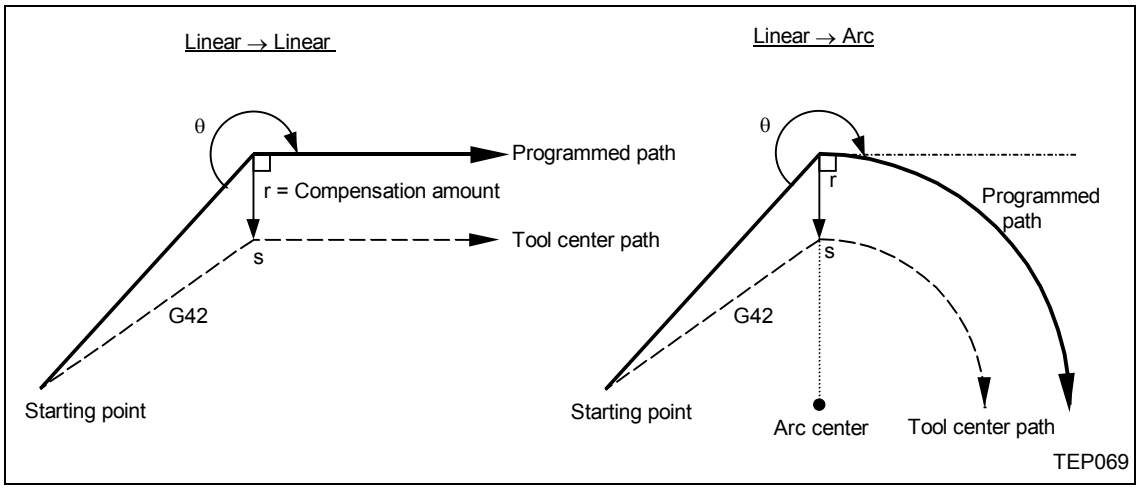


Fig. 2-19

B. For the corner exterior (obtuse angle) [90° ≤ θ <180°]

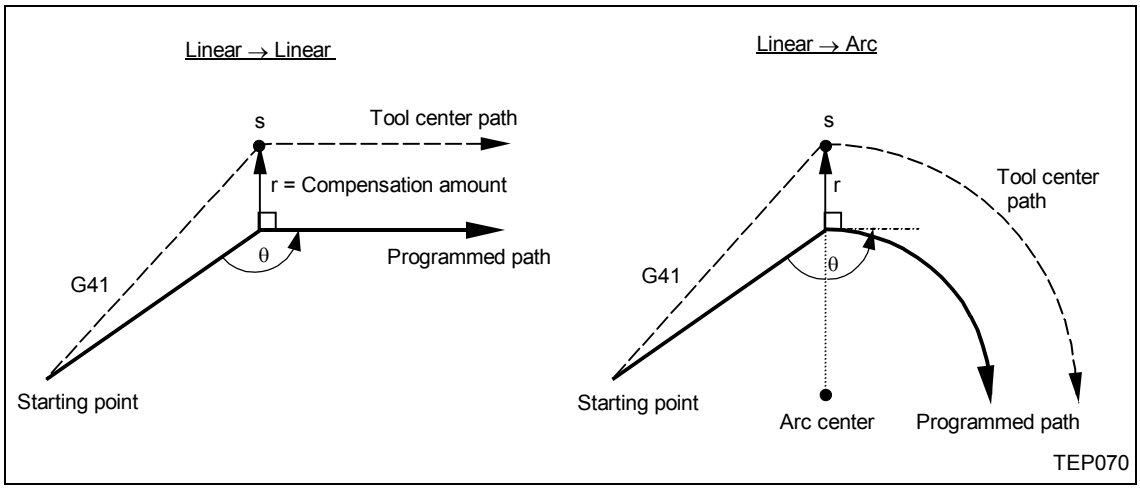


Fig. 2-20

C. For the corner exterior (acute angle) [θ < 90°]

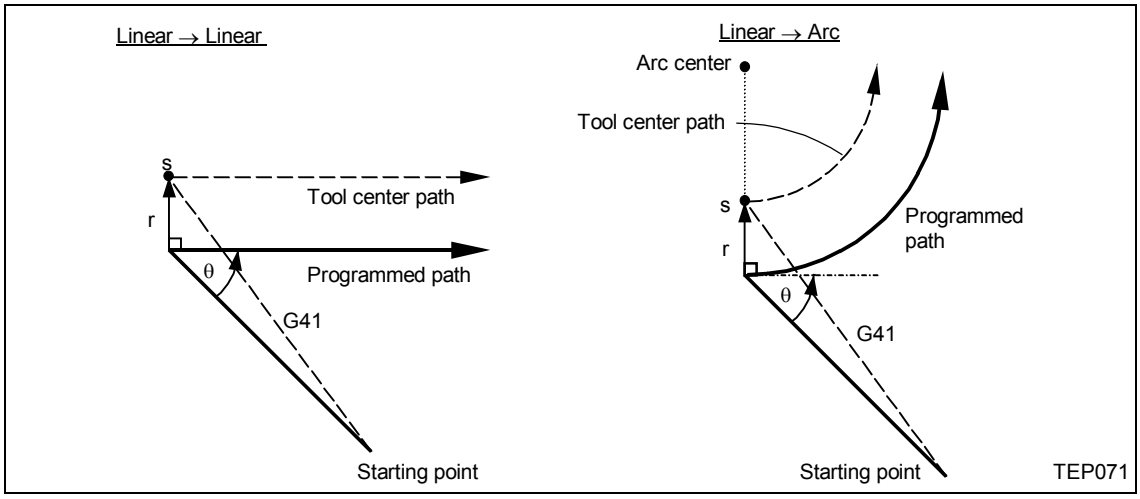


Fig. 2-21

4. Operation during the compensation mode

Compensation is performed for linear or circular interpolation as well as positioning commands. Identical compensation command G41 or G42 will be ignored if it is used in the compensation mode.

Successive setting of four or more blocks that do not contain axis movement command in the compensation mode will result in excessive or insufficient cutting.

A. For the corner exterior

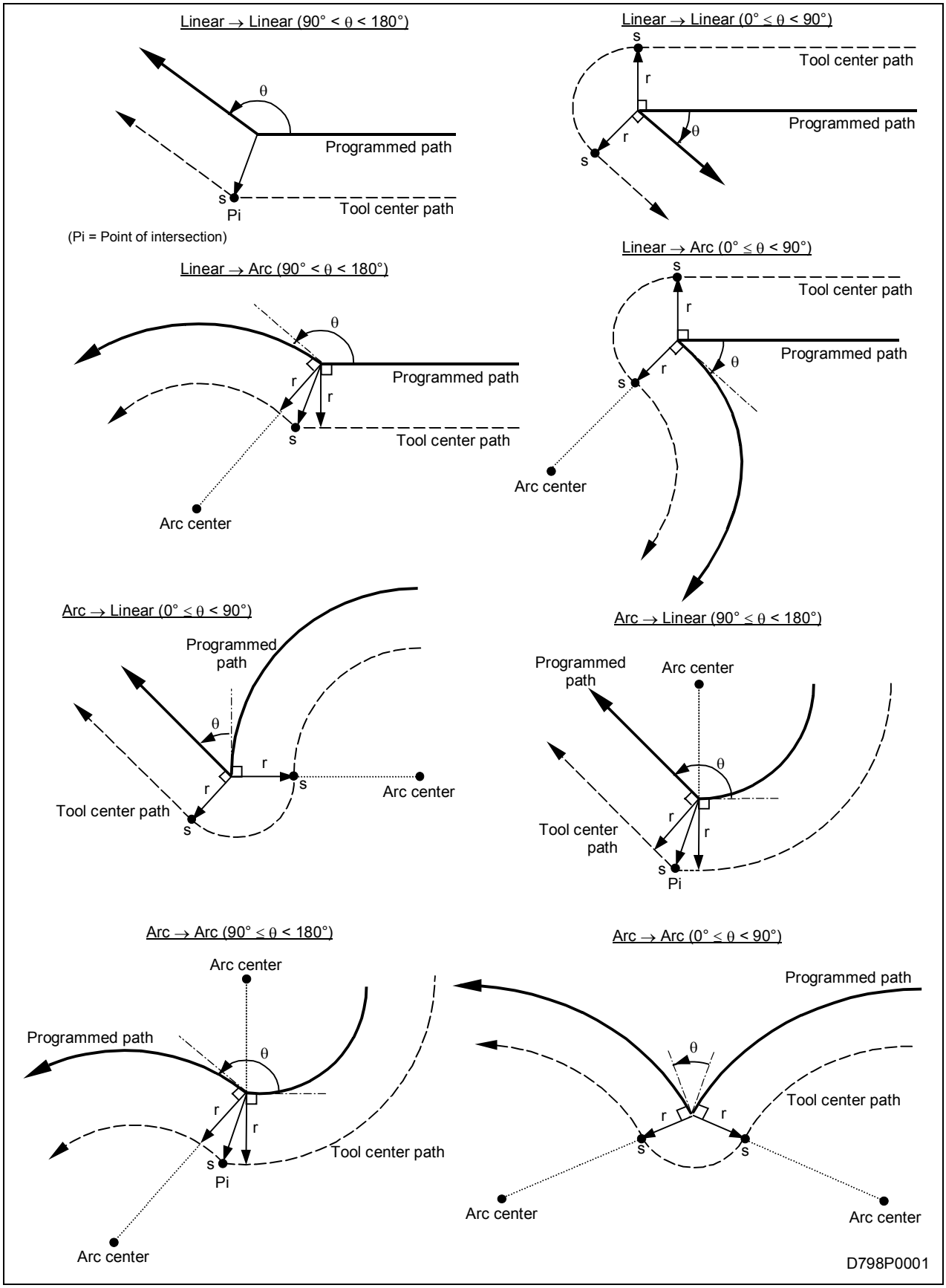


Fig. 2-22

B. For the corner interior

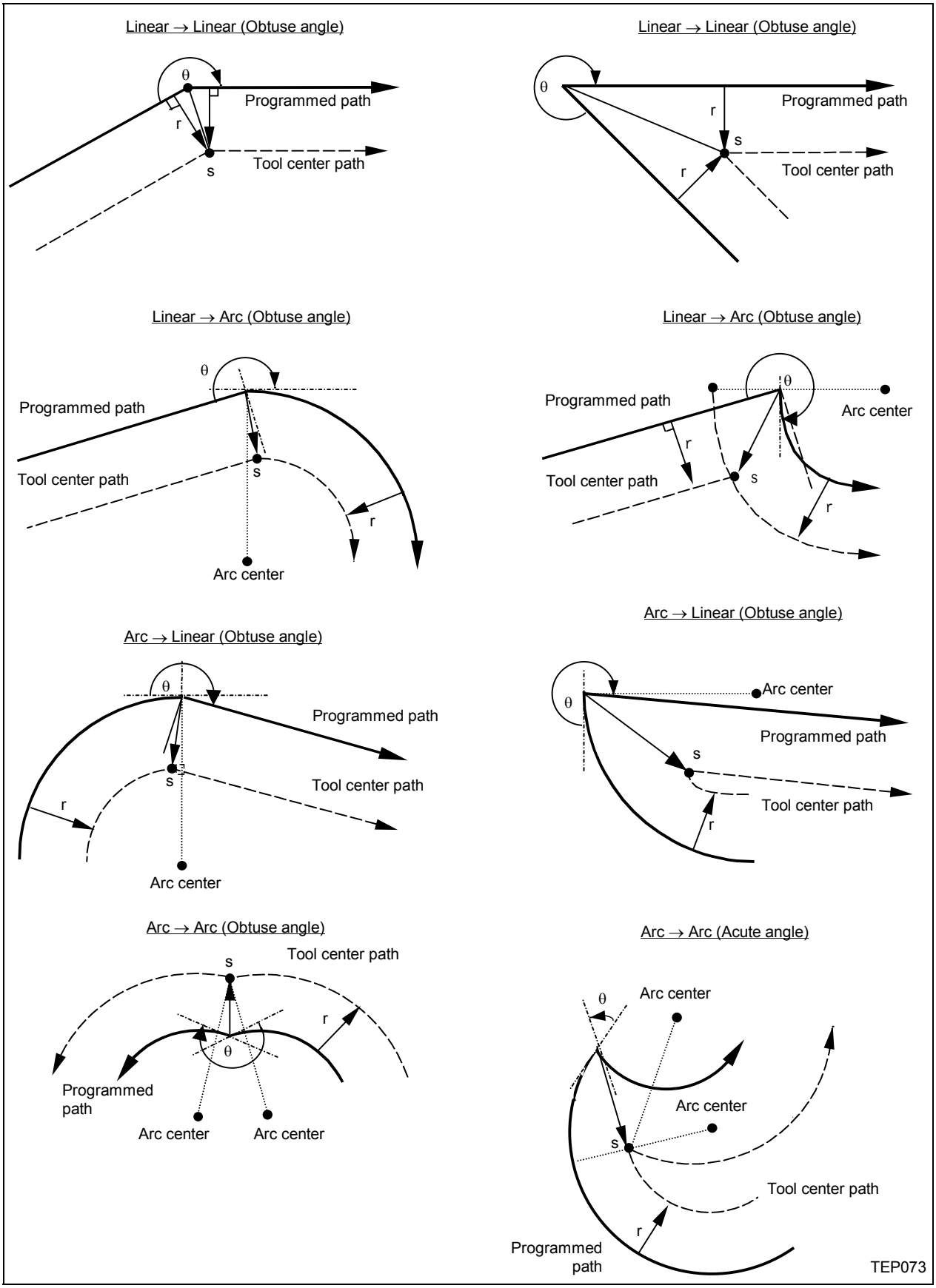


Fig. 2-23

5. Cancelling tool radius compensation

The mode of tool radius compensation will be cancelled in any of the two cases listed below.

- Command G40 has been executed.
- Command for compensation data number “zero” (D00) has been executed.

The command used with the compensation cancellation command must be a move command other than those for circular interpolation. An alarm will occur if an attempt is made to cancel the compensation mode with a circular interpolation command.

When the compensation cancellation command is read into the compensation buffer, the cancellation mode is set and subsequent blocks of data are read directly into the pre-read buffer, not into the compensation buffer.

6. Cancelling operation for tool radius compensation

A. For the corner interior

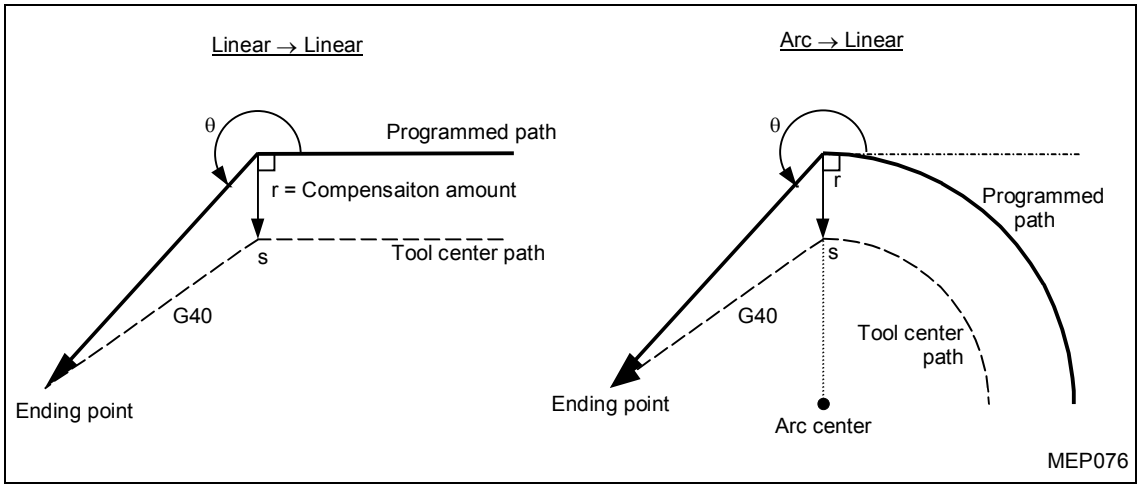


Fig. 2-24

B. For the corner exterior (obtuse angle)

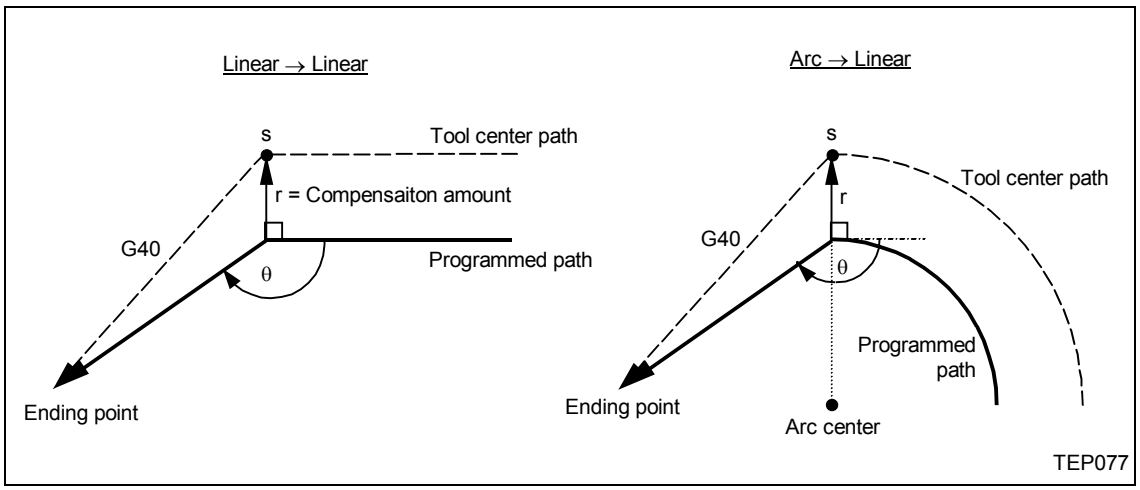


Fig. 2-25

C. For the corner exterior (acute angle)

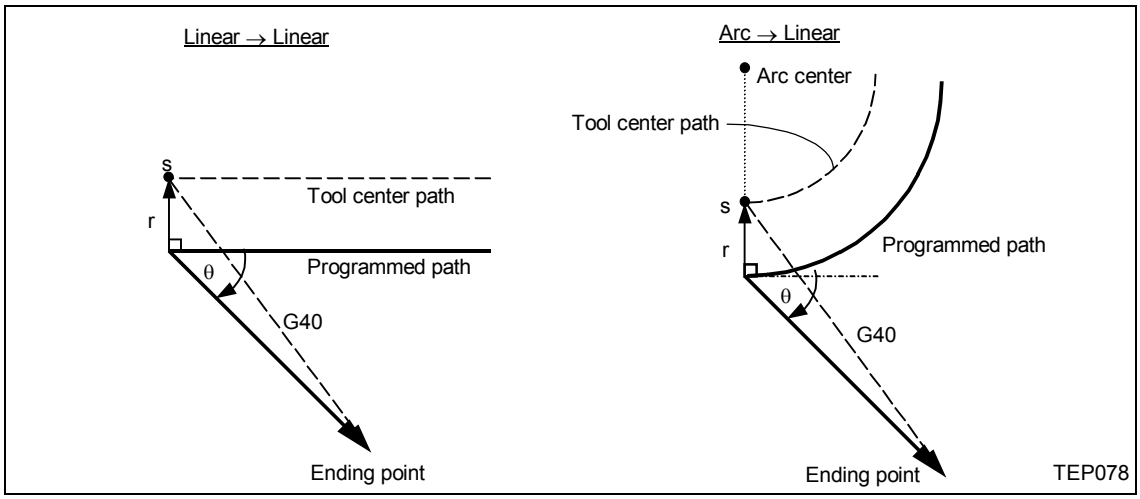


Fig. 2-26

7. Interference check for tool radius compensation

The interference here refers to an overcutting of the workpiece as a result of tool radius compensation, and the function for automatically preventing such interference is referred to as interference check. The following shows various patterns of the interference in question, together with their preventive actions.

Note: The function of interference check for tool radius compensation is an improvement introduced in the CNC, version v59.07.

A section of foreseen interference is marked with a red star, as shown below, along with an alarm message (**303403 Interference**) on the screen. The occurrence of this alarm, however, does not cause the automatic operation to be stopped.

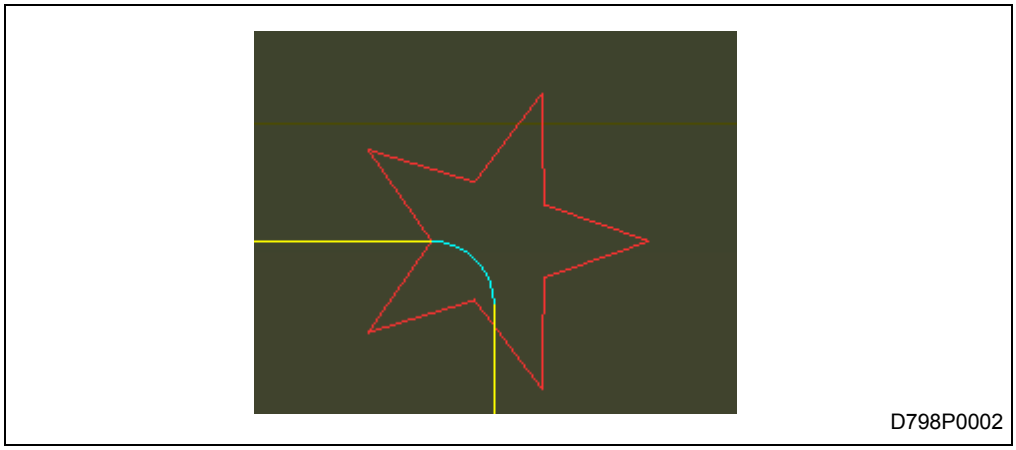


Fig. 2-27

A. Interference patterns and preventive actions

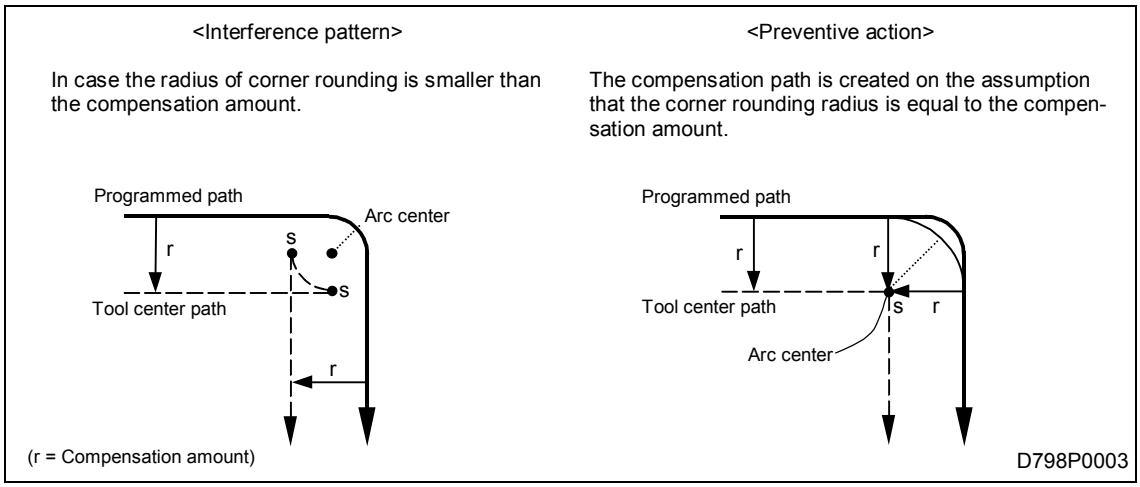


Fig. 2-28

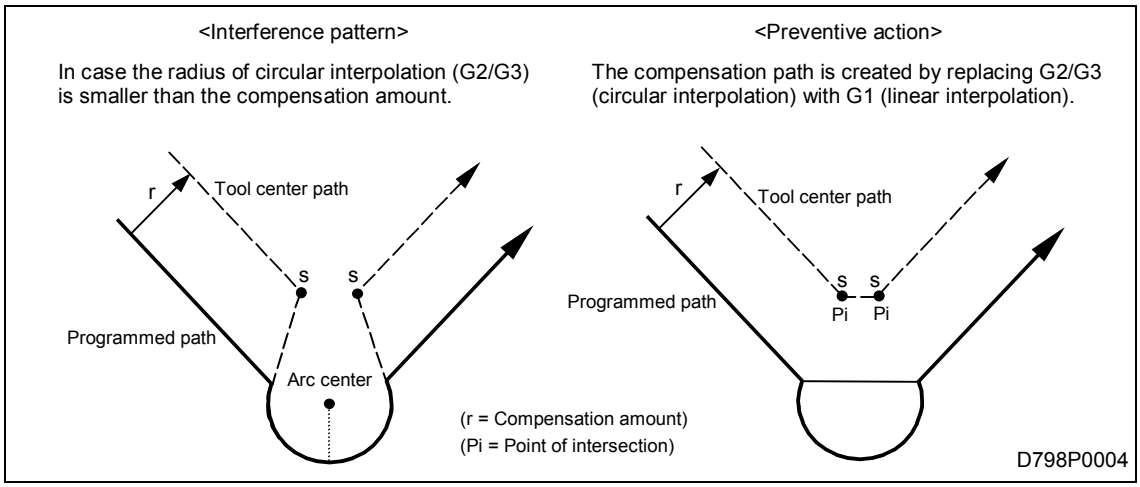


Fig. 2-29

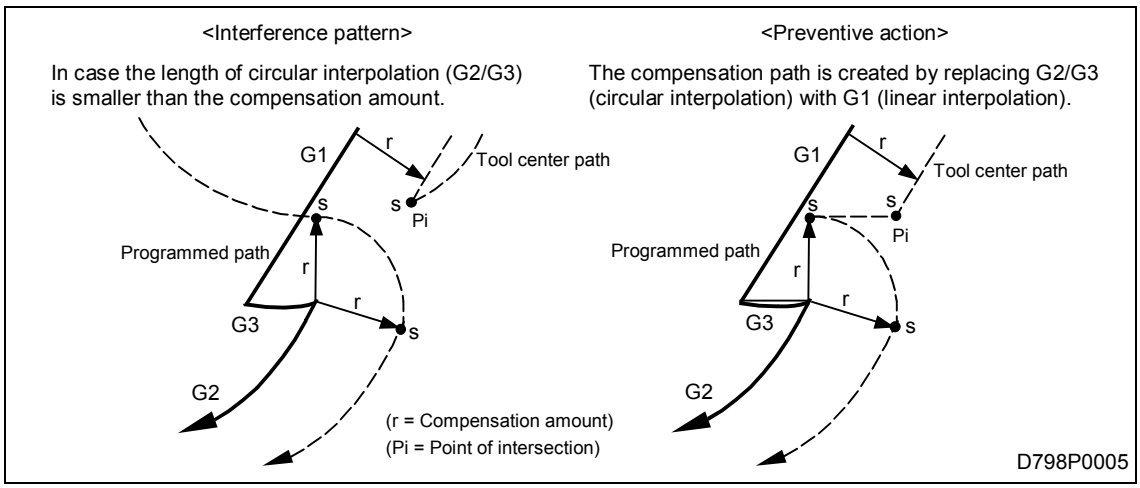


Fig. 2-30

<Interference pattern>

The size of recess is so small with respect to the compensation amount that the compensation path should include some segments the direction of which is opposite to that of the programmed path.

<Preventive action>

The compensation path is created by skipping the troublesome blocks.

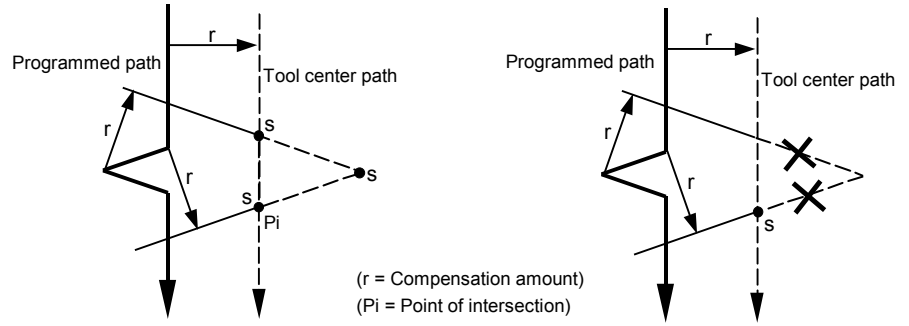


Fig. 2-31

<Interference pattern>

The size of recess is so small with respect to the compensation amount that the compensation path should include some segments the direction of which is opposite to that of the programmed path.

<Preventive action>

The compensation path is created by skipping the troublesome blocks.

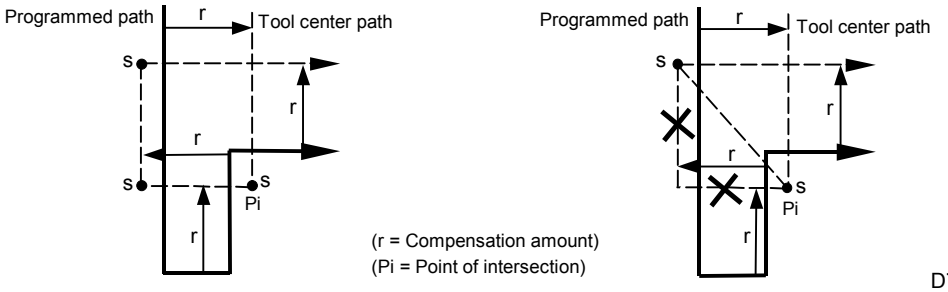


Fig. 2-32

The example below is given to show a case where, if the troublesome blocks are unconditionally skipped to prevent reverse-motion segments from being included, the center of a circular section on the compensation path should be shifted from that of the original one on the programmed path.

To cope with this problem, the skipping is suppressed for a block in question if it is followed by a block of circular interpolation.

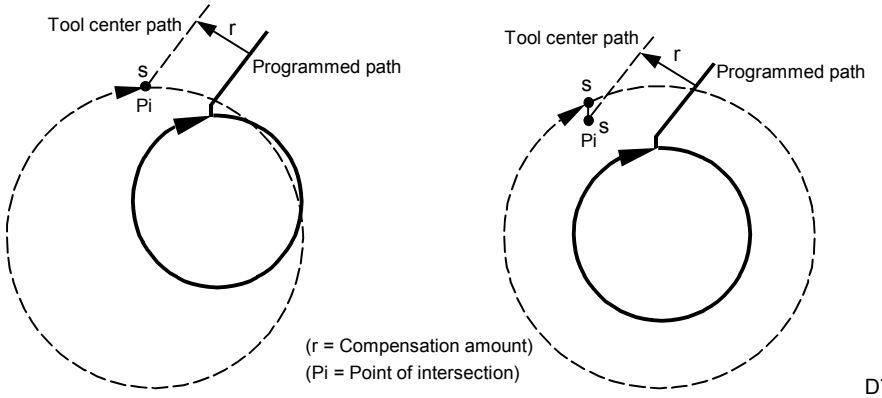


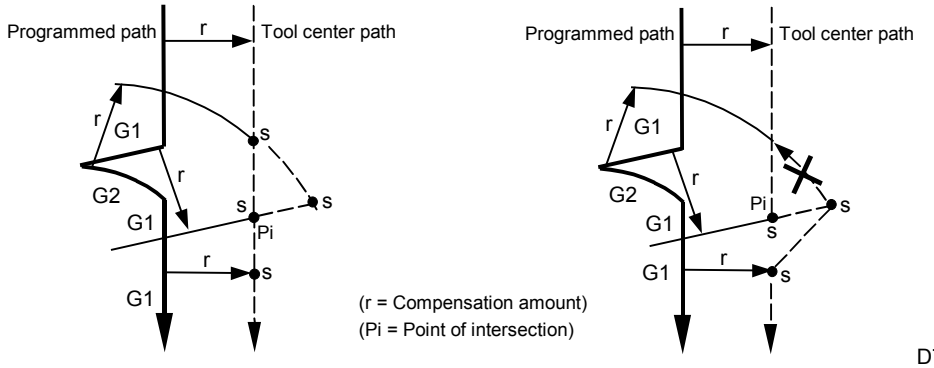
Fig. 2-33

<Interference pattern>

The size of recess is so small with respect to the compensation amount that the compensation path should include some segments the direction of which is opposite to that of the programmed path.

<Preventive action>

The compensation path is created by skipping the troublesome blocks on condition that their succeeding block is of linear (not of circular) interpolation.



D798P0009

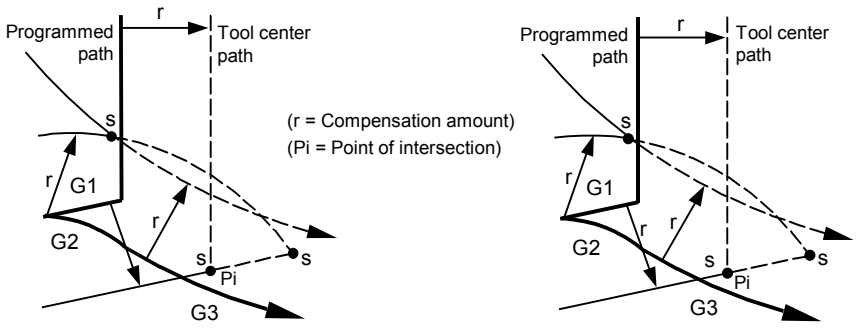
Fig. 2-34

<Interference pattern>

The size of recess is so small with respect to the compensation amount that the compensation path should include some segments the direction of which is opposite to that of the programmed path.

<Preventive action>

The compensation path is created by skipping the troublesome blocks on condition that their succeeding block is of linear (not of circular) interpolation.

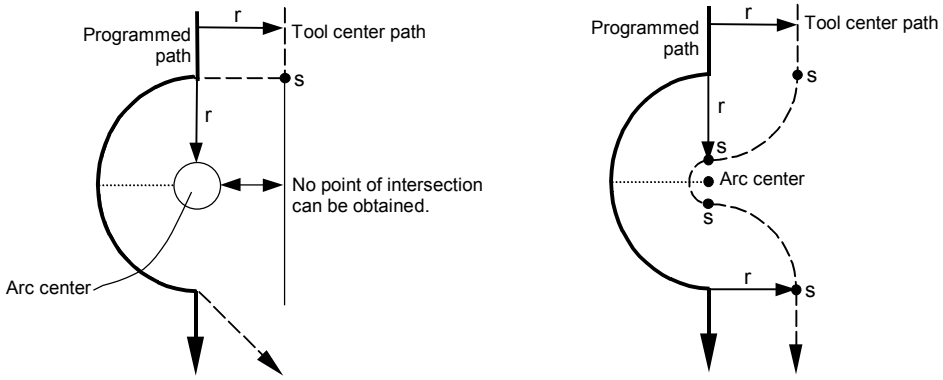


D798P0010

Fig. 2-35

The example below is given to show a case where the point of intersection cannot be obtained for tool radius compensation due to too large a compensation amount with respect to the size of the geometric figures.

In such cases a circular element will be interpolated to create a continuous compensation path.



D798P0011

Fig. 2-36

B. Related parameters

Use the following parameter to select the type of processing for the function of interference check during tool radius compensation:

User parameter No.	Function	Conditions of effectiveness	Setting
N093	Type of processing for the foreseen interference during tool radius compensation	Immediately	0 : Automatic prevention of interference 1 : Operation stop with an alarm

Set **N093** to 0 to continue the automatic operation by automatic creation of compensation path for preventing the foreseen interference during tool radius compensation.
Set **N093** to 1 to stop the automatic operation with an alarm (**709 OFFSET INTERFERENCE**) for any foreseen interference during tool radius compensation.

2-10-3 Programmed compensation data input: G10

1. Function and purpose

The function code G10 is provided to set or modify, in the flow of the program, the radius compensation amounts stored on the **BEAM OFFSET** screen under D1 to D64.

2. Programming format

```
G10 Dd Rr;
```

d: Compensation data number (1 to 64)

r: Compensation amount
Setting range: 0.000 to 9.999 mm or 0.0000 to 0.3936 in.

3. Detailed description

A block of the above format changes the stored compensation amount according to the argument R. The value “r” is used for a replacement and an additive modification, respectively, when the G10 command is given in the absolute programming mode (G90) and the incremental programming mode (G91). The new value can then be fetched in the succeeding blocks with the corresponding compensation number (D-code).

Example:
```
G10 D1 R5.000;
G42 D1 G01 X20. F1000;
```

2-11 Programming Support Functions

2-11-1 Subprogram control commands: M98, M99

1. Function and purpose

Prestoring a fixed sequence or pattern to be repeatedly used into the memory as a subprogram allows the subprogram to be called from the main program as required.
Use command M98 to call a subprogram, and use command M99 to return from the subprogram. You can also call another subprogram from the current one. Its maximum available depth of nesting (call of one subprogram from another) is of nine levels.

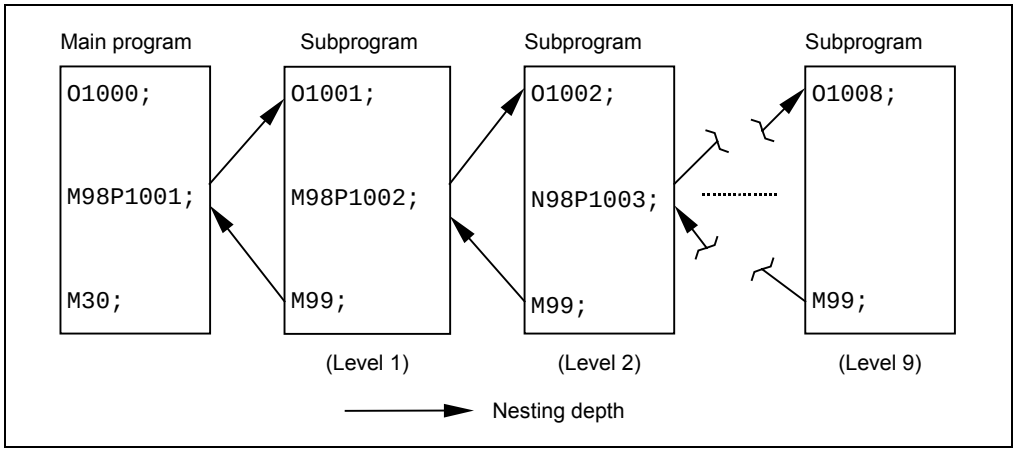


Fig. 2-37

2. Programming format

Subprogram call

```
M98 P_L_;
```

Subprogram execution repeat times (If omitted, then L1)
Program number of the subprogram to be called

Return from the subprogram

```
M99;
```

Return to the block following the call block

3. Creating and registering subprograms

The subprogramming format is similar to the usual machining program creation format for memory operation, except that the subprogram termination instruction M99 must be included in the final block as a single-command block.

```
O□□□□;
 ..............;
 ..............;
  ⋮                          } Subprogram
 ..............;
M99;            Subprogram termination
%               Registration complete code
```

Prepare programs of the above format and store them in the NC memory.

With the exception of those predetermined for special use, any number from 1 to 9999 can be used for subprogram identification. The maximum available depth of subprogram nesting is of nine levels. An alarm will occur if this limit is exceeded.

Main and subprograms are stored into the memory without distinction under the specified number. The identification numbers of the main and subprograms stored, therefore, must not overlap; otherwise an alarm will occur during registration.

4. Executing a subprogram

M98: Subprogram call command
M99: Subprogram termination command

Programming format

```
M98 Pp Lℓ;
```

p : Number of the subprogram to be called (in max. four digits)
ℓ : Repeat times (in max. four digits, from 1 to 9999)
Execution will occur only once if L is omitted.
For example,

```
M98 P1 L3; is equivalent to:
M98 P1;
M98 P1;
M98 P1;
```

Example: For three subprogram calls (level three nesting):

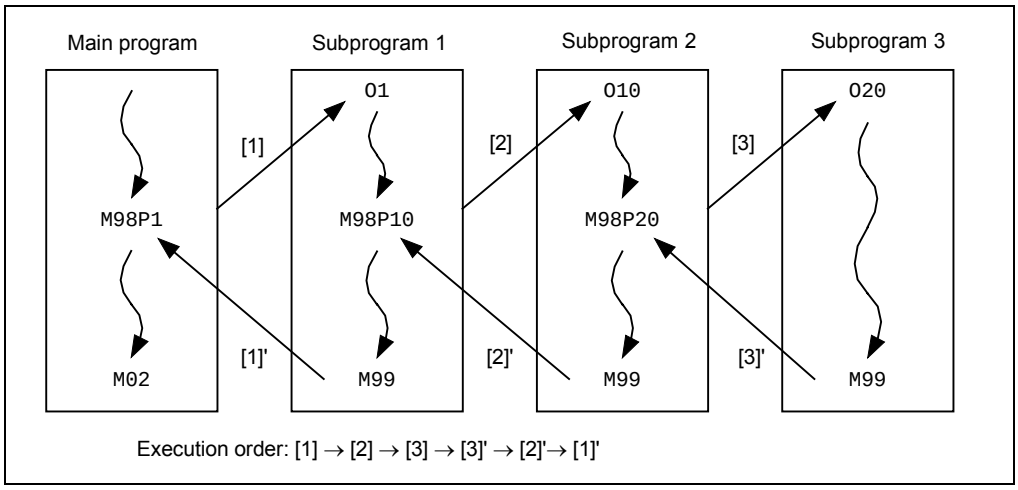


Fig. 2-38

For nesting, the codes M98 and M99 must be given in a one-to-one correspondence (for example [1] for [1]', [2] for [2]', and so on up to the final M98 or M99).
Modal information is automatically updated in the order of actual execution, irrespective of whether the program is a main program or a subprogram. After subprogram call, therefore, extra careful attention must be paid to the status of the modal data.

5. Miscellaneous notes

1. An alarm will result if a program of the number specified by P is not found.
2. Setting an M99 code in the main program returns the program flow to the starting block.

2-11-2 Programmed coordinate system rotation: G68, G69

1. Function and purpose

Programmed coordinate system rotation is performed to rotate the machining shape itself on the workpiece by rotating the workpiece coordinate system.

2. Programming format

`G68 Xx Yy Rr;` ..... Coordinate system rotation

x, y: X and Y coordinates of the center of rotation

r: Angle of rotation (for counterclockwise rotation, use the plus sign)
Set an angle from –360.000 to +360.000 degrees in 0.001 degree steps.

`G69;` ..... Cancellation of coordinate system rotation
This command code can be included in any block, whether independent from or together with other command codes.

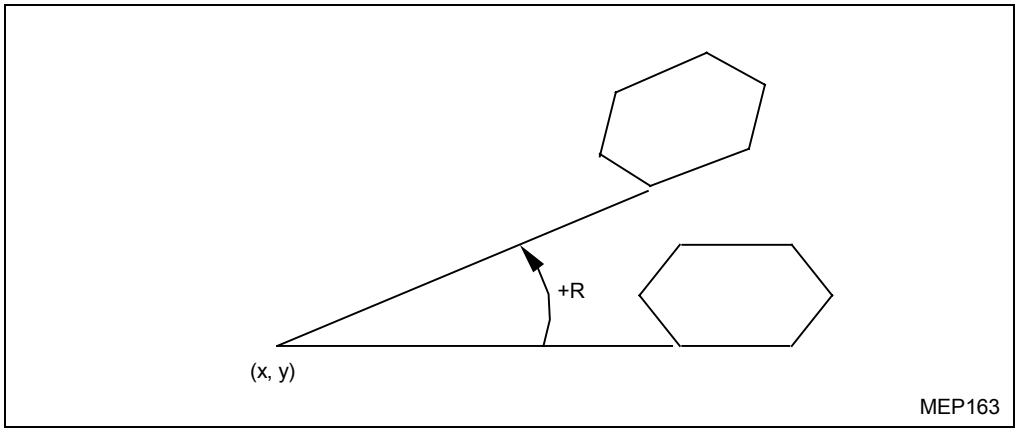


Fig. 2-39

3. Detailed description

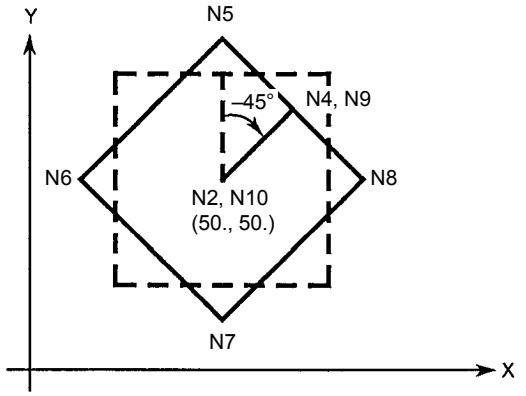
1. The arguments X and Y for the center of rotation can be omitted when the rotation is to be done around the current position (taken before giving the G68 command).
2. The arguments X and Y for the center of rotation must always be entered by absolute data input, but the argument R for the angle of rotation is to be entered in absolute or incremental data according to the curret modal condition of G90 or G91.
3. A new command of G68 in the mode of coordinate system rotaiton will be processed for changing the center, and the angle, of rotation.

4. Correlationships to other functions, and precautions

1. Coordinate system rotation is valid only for the automatic operation; it is invalid for manual cutting feed, rapid traverse, and handle pulse feed. Moreover, it is invalid for automatic or manual return to the reference point (zero point).
2. Tool offset compensation is processed lastly. That is, coordinate system rotation is performed before tool radius compensation.
3. The digital counter of the current position represents the movements on the rotated coordinate system. If coordinate system rotation is carried out for a single-axis command, therefore, simultaneous two-axis movements will be displayed in general.

5. Programming example

```
N1   G92 X0 Y0;
N2   G90 G00 X50.Y50.;
N3   G68 R-45.;
N4   G01 Y75.F1000;
N5   X25.;
N6   Y25.;
N7   X75.;
N8   Y75.;
N9   X50.;
N10  G69 G00 X50.Y50.;
N11  M30;
```



LP111-00121

Fig. 2-40

2-11-3 Scaling: G50, G51

1. Function and purpose

The shape specified in a machining program can be enlarged or reduced in size using a scaling command with G51.

Use a G51 command to specify a scaling axis, the center of scaling, and a scaling factor.
Use a G50 command to cancel the scaling function.

2. Programming formats

`G51 Xx Yy Pp;`	Scaling ON (specify a scaling axis, the center of scaling (in incremental or absolute data), and a scaling factor)
`G50;`	Scaling OFF

3. Detailed description

A. Selection of scaling

The scaling mode is selected by a G51 command. The block of G51 is used to specify a scaling axis, the center of scaling, and a scaling factor, and does not cause any axis motion.
Scaling becomes valid only for the axis to which the center of scaling has been specified.

- Center of scaling
  The center of scaling must be specified with the axis addresses according to the absolute or incremental data input mode (G90 or G91).
  Simply omit the arguments X and Y to set the current position as the center.

- Scaling factor
  Use address P (in the same block as G51) to specify a scaling factor.
  Specifiable range of factors: 1 to 9999 (for 0.001 to 9.999 times)

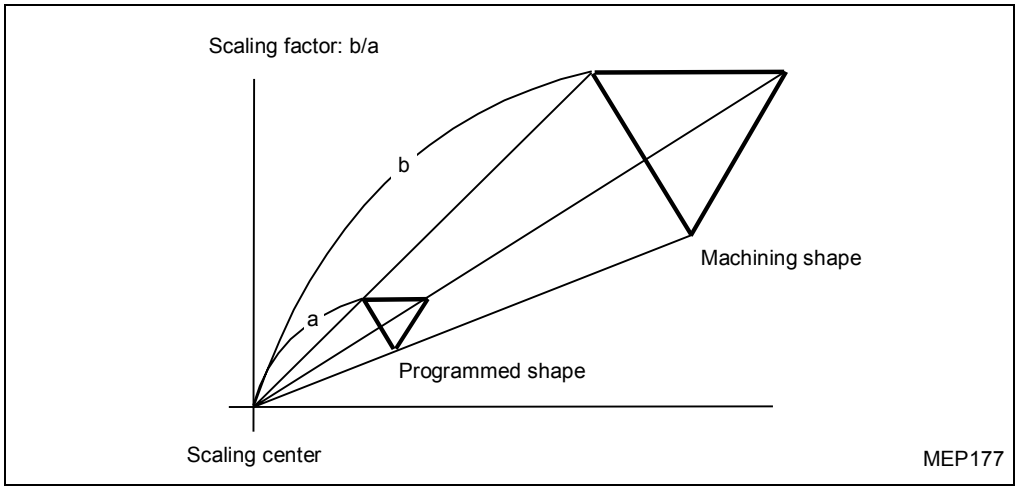


Fig. 2-41

B. Cancellation of scaling

The scaling is cancelled by a G50 command. This command cancels any deviation between the programmed position and the actual tool position.

C. Precautions

1. Scaling does not affect the tool radius compensation in any way. The compensation is performed on the ‘scaled’ shape.
2. Scaling is valid only for move commands in the automatic mode of operation; it is not valid for manual movement.
3. After-scaling coordinates are displayed as position data.
4. Scaling is performed on the axis for which the center of scaling is specified by G51. In that case, scaling becomes valid for all move commands in the automatic mode of operation.
5. Scaling will be cancelled if M02 or M30 is executed. Scaling is also cancelled by pressing the reset key.
6. The center of scaling is shifted accordingly if the coordinate system is shifted using commands G92, G54 to G59, or G52 during scaling.

4. Programming example

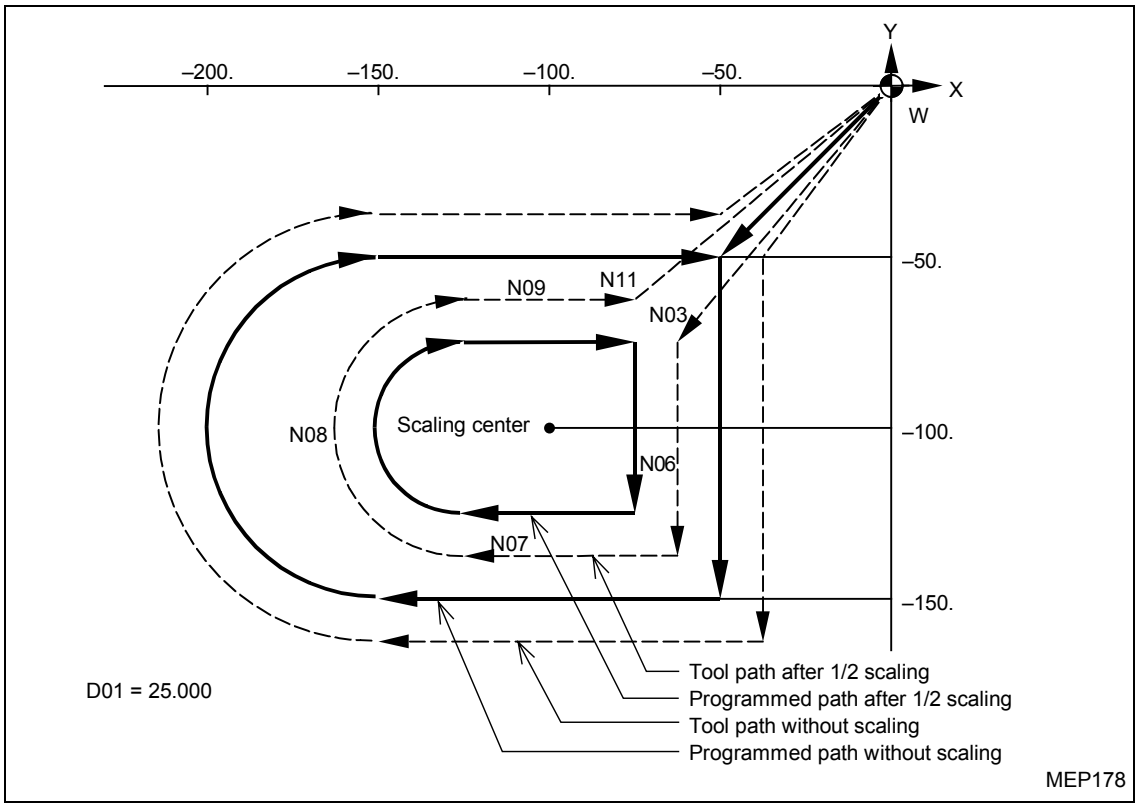


Fig. 2-42

```
N01  G92      X0. Y0;
N02  G90  G51 X-100.Y-100.P500;
N03  G41  G00 X-50.Y-50.D01;
N06  G01      Y-150.F1000;
N07           X-150.;
N08  G02      Y-50.J50.;
N09  G01      X-50.;
N11  G40  G00 G50 X0Y0;
N12  M02;
```

2-11-4 Programmable mirror image: G50.1, G51.1

1. Function and purpose

You can use G-codes to turn the mirror image mode on or off for each axis.

2. Programming format:

```
G51.1 Xx Yy;    Mirror image ON
G50.1 Xx Yy;    Mirror image OFF
```

3. Detailed description

1. When using command G51.1, the name of the axis for which mirror image processing is to be performed must be designated using the appropriate coordinate word, and the mirror image center coordinates must be designated using absolute or incremental data.
2. If the coordinate word is designated in G50.1, then this denotes the axis for which the mirror image is to be cancelled. Coordinate data is ignored in that case.
3. After mirror image processing has been performed for only one of the axes X and Y, the rotational direction and the offset direction become reverse during circular interpolation, tool radius compensation, or coordinate system rotation.
4. Since the mirror image processing function is valid only for local coordinate systems, the center of mirror image processing moves according to the particular workpiece coordinate offsetting data.

4. Programming example

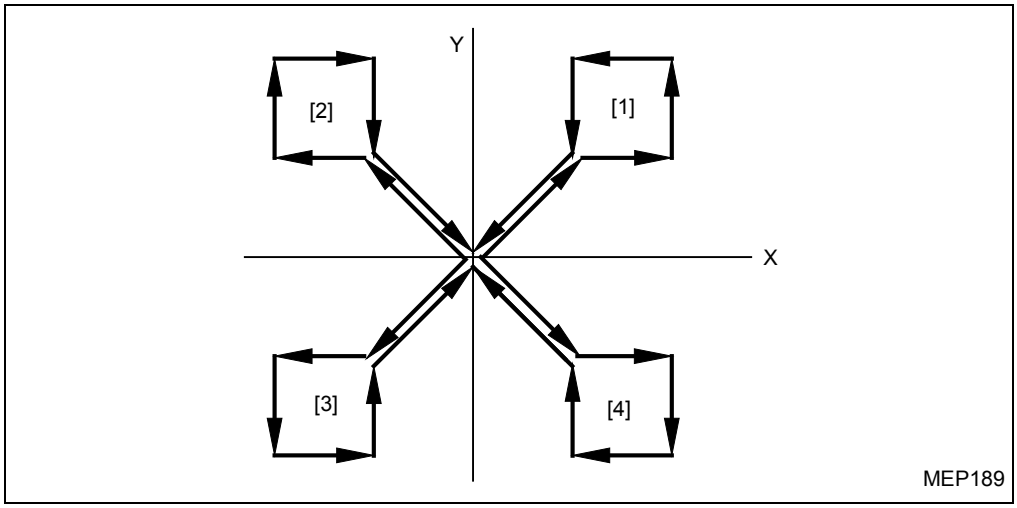


Fig. 2-43

```
                          X     Y
M98 P100; ——— [1]   OFF   OFF
G51.1X0;  ┐
M98 P100; ┘—— [2]   ON    OFF
G51.1Y0;  ┐
M98 P100; ┘—— [3]   ON    ON
G50.1X0;  ┐
M98 P100; ┘—— [4]   OFF   ON
G50.1X0Y0;          OFF   OFF
```

2-11-5 Corner chamfering and rounding

Any corner between successive motion blocks can be chamfered and rounded easily by adding arguments #C and #R respectively to the first block.

1. Corner chamfering (#C)

A. Function

The virtual corner between two straight lines is chamfered on a line running through those two points which are obtained by shortening both sides from the virtual intersection by the dimension specified with #C.
For a circular path the above-mentioned shortening is performed along the arc.

B. Programming format

```
G01 X_ Y_ #Cc1;
G01 X_ Y_;
```

The corner between two successive linear-motion blocks is chamfered so that the starting and ending points of chamfering may be distant from the virtual corner by "c_1".

Similar format of the first block applies for a circular-motion block:

```
G02 (or G03) X_ Y_ I_ J_ #Cc1;  or
G02 (or G03) X_ Y_ R_ #Cc1;
```

C. Remarks

1. The starting point for the second block of corner chamfering is the virtual corner.
2. A programming error occurs if the character C is not preceded by #.
3. If both commands #C and #R are given in the same block, the command set at last will be performed.
4. Tool radius compensation is performed on the corner-chamfered shape.
5. Scaling function also works on the corner chamfering.
6. A programming error occurs if the second block of corner chamfering contains no interpolation command (of G01, G02 and G03).
7. A programming error occurs if the motion distance of the first block of corner chamfering is not larger than the chamfering size (c_1).
8. A programming error occurs if the motion distance of the second block of corner chamfering is not larger than the chamfering size (c_1).

D. Example

```
[1] G91 G01 X100.#C10.;
[2] X100.Y100.;
```

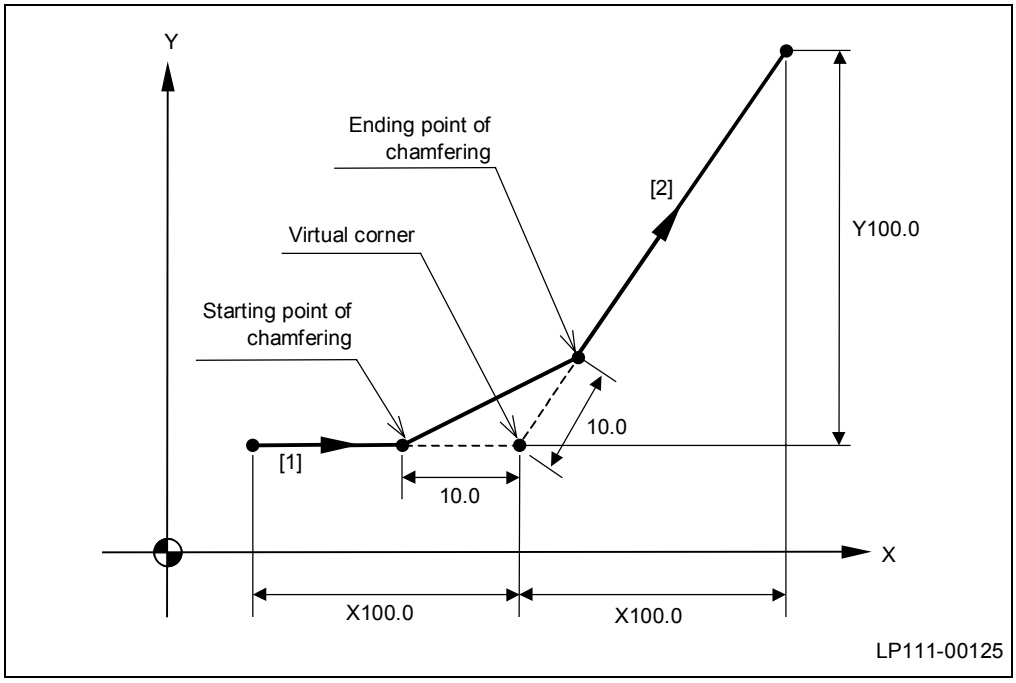


Fig. 2-44

```
[1] G90 G03 X60.Y50.I0 J40.#C20.;
[2] G01 X100 Y20.;
```

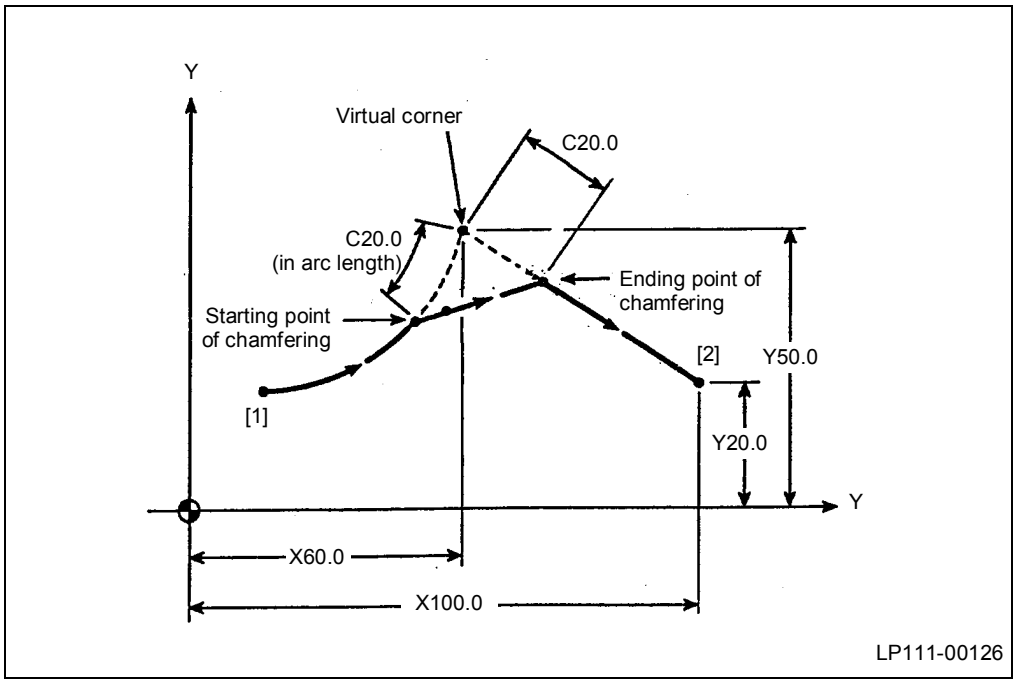


Fig. 2-45

2. Corner rounding (#R)

A. Function

The virtual corner between two motion blocks can be rounded by the interpolation of an osculatory circular arc which has a radius of the dimension designated with #R.

B. Programming format

```
G01 X_ Y_ #Rr1;
G01 X_ Y_;
```

The corner between two successive linear-motion blocks is rounded by an interpolated osculatory arc of the radius "r_1".

Similar format of the first block applies for a circular-motion block:

```
G02 (or G03) X_ Y_ I_ J_ #Rr1;  or
G02 (or G03) X_ Y_ Rr #Rr1;
```

where "r" denotes the radius for the circular interpolation.

C. Remarks

1. The starting point for the second block of corner rounding is the virtual corner.
2. A programming error occurs if the character R is not preceded by #.
3. If both commands #C and #R are given in the same block, the command set at last will be performed.
4. Tool radius compensation is performed on the corner-rounded shape.
5. Scaling function also works on the corner rounding.
6. A programming error occurs if the second block of corner rounding contains no interpolation command (of G01, G02 and G03).
7. A programming error occurs if the motion distance of the first block of corner rounding is not larger than the rounding size (r_1).
8. A programming error occurs if the motion distance of the second block of corner rounding is not larger than the rounding size (r_1).

D. Example

```
[1] G91 G01 X100.#R10.;
[2] X100.Y100.;
```

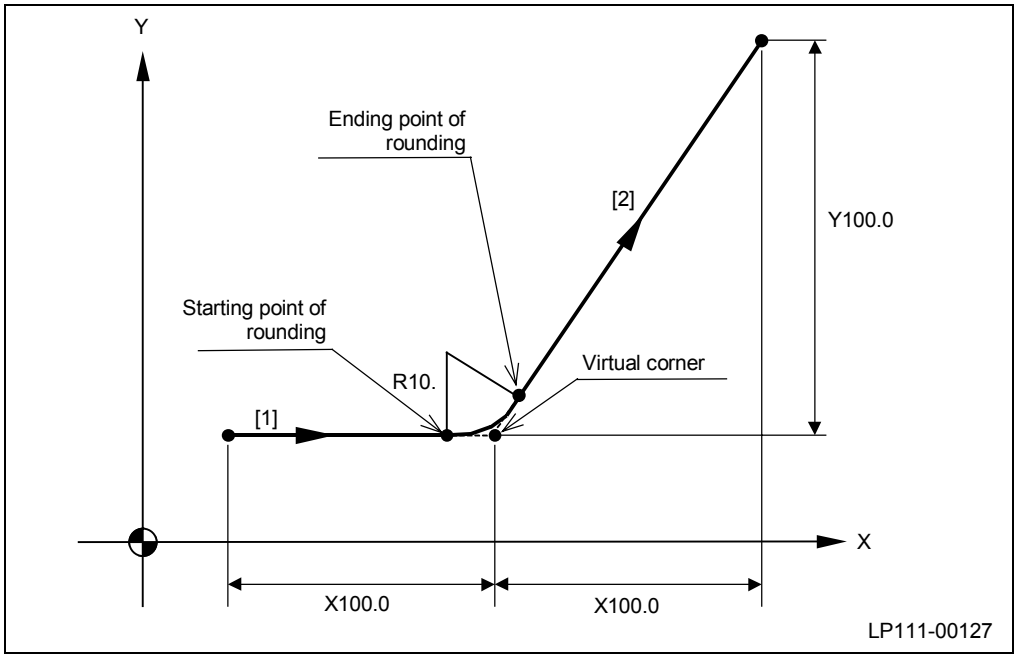


Fig. 2-46

```
[1] G90 G03 X100.Y100.I–10.J60.#R20.;
[2] G01 X180 Y80.;
```

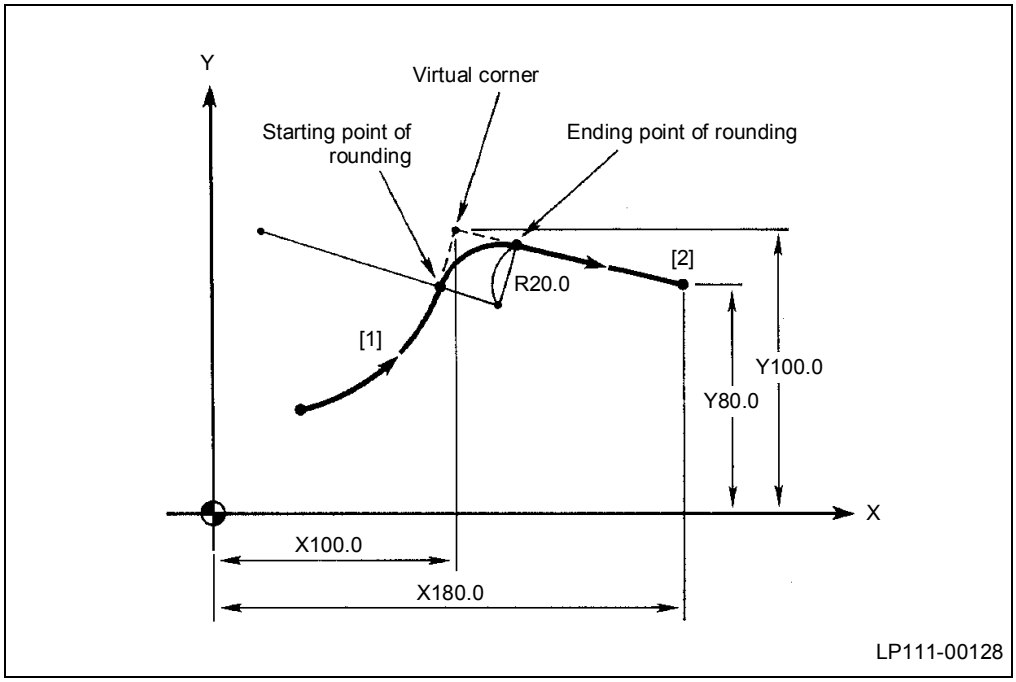


Fig. 2-47

2-11-6 Polar coordinates input: G16, G15

1. Function and purpose

The ending point of axis movement can be specified in polar coordinates (radius and angle).

2. Programming format

`G16 X_ Y_;` Polar coordinates input

Use the addresses X and Y respectively to specify the radius and the angle.

`G15;` Polar coordinates input OFF

3. Detailed description

1. Positive values of angle refer to counterclockwise measurement from the positive direction of the X-axis.
2. Absolute and incremental programming modes (G90 and G91) are valid for the specification of the radius and the angle.

   - For absolute programming of the radius
     Take the origin of the local coordinate system as the center of polar coordinates.

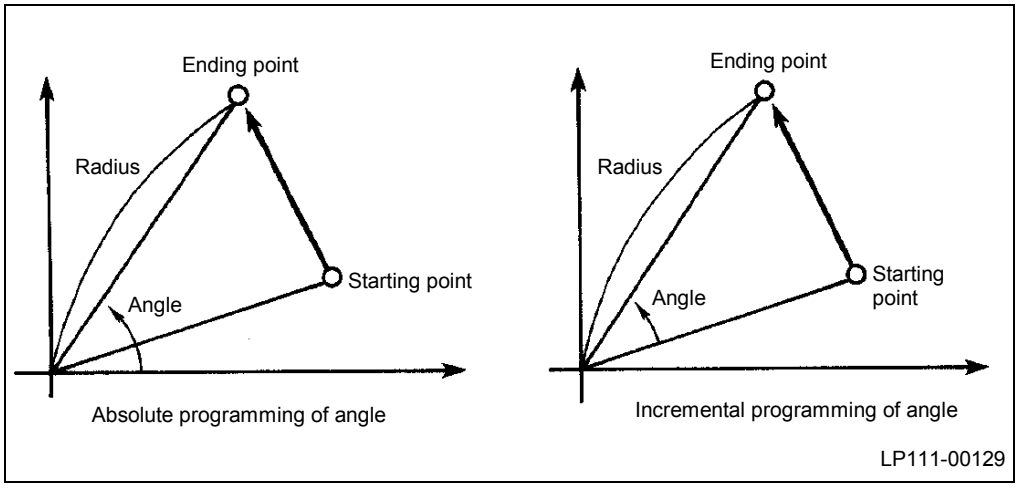


Fig. 2-48

   - For incremental programming of the radius
     Take the current position (starting point) as the center of polar coordinates.

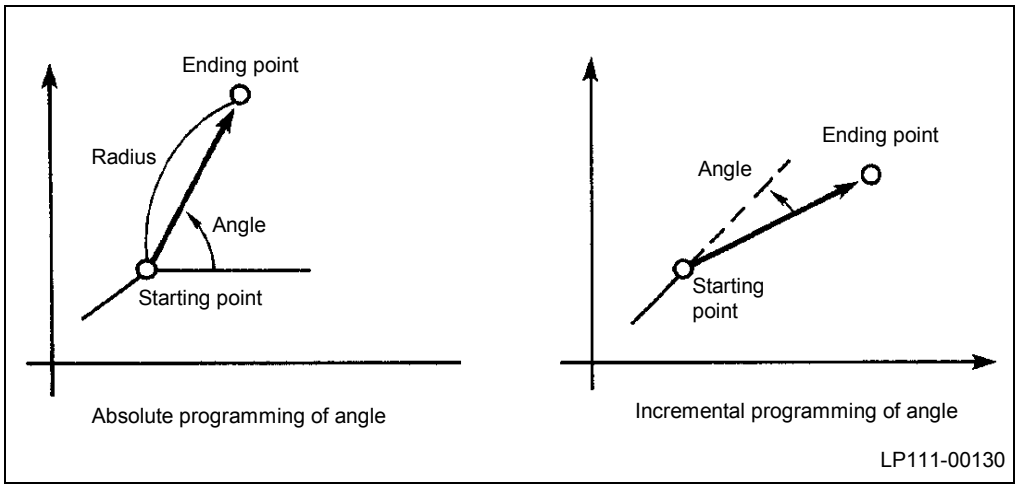


Fig. 2-49

3. Arguments X and Y under the functions shown below are properly excluded from the processing for polar coordinates input.

   Dwell (G04)
   Local coordinate system setting (G52)
   Coordinate system setting (G92)
   Programmed coordinate system rotation (G68)
   Programmable mirror image (G50.1)

4. G15 may be given in a block with other commands.

4. Programming example

1. Absolute programming of both radius and angle

```
N1  G92 X0 Y0;
N2  G90 G16;
N3  G01 X30. Y30. F1000;
N4  Y150.;
N5  Y270.;
N6  G15 X0 Y0;
N7  M30;
```

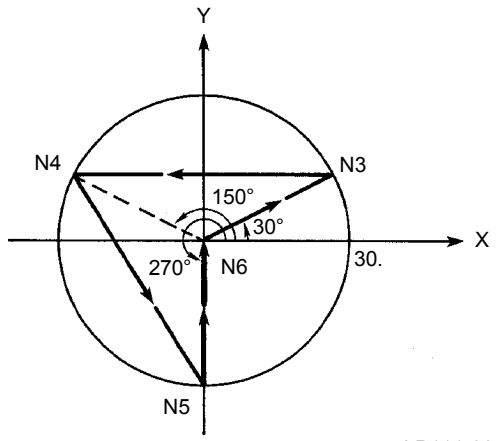


Fig. 2-50

2. Absolute programming of radius and incremental programming of angle

```
N1  G92 X0 Y0;
N2  G90 G16;
N3  G01 X30. Y30. F1000;
N4  G91 Y120.;
N5  Y120.;
N6  G15 G90 X0 Y0;
N7  M30;
```

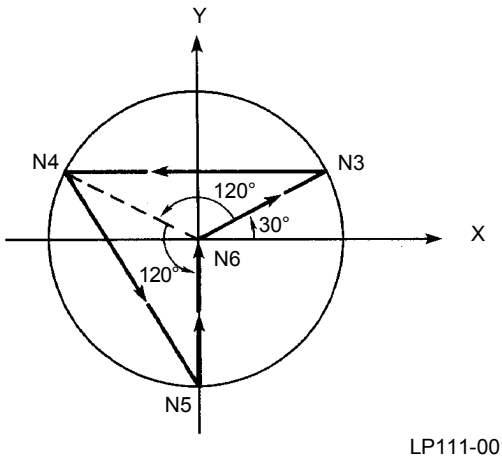


Fig. 2-51

2-12 Coordinate System Setting Functions

2-12-1 Machine coordinate system, workpiece coordinate systems, and local coordinate systems

The machine coordinate system, which is a fixed coordinate system for the machine, is used to designate the specific positions to the machine.
Workpiece coordinate systems are the coordinate systems to be used by a programmer when creating programs. These coordinate systems have their origins set at reference points on workpieces. Local coordinate systems are the coordinate systems to be set on the workpiece coordinate systems to facilitate the creating of partial-machining programs.

2-12-2 Machine zero point and second, third, and fourth reference points

The machine zero point refers to a position that acts as a reference point for the machine coordinate system. That is, the machine zero point is a point specific to the machine, determined by return to the reference point (zero point).

The second, third, and fourth reference points (zero point) are the points whose positions in the machine coordinate system are parameter-preset.

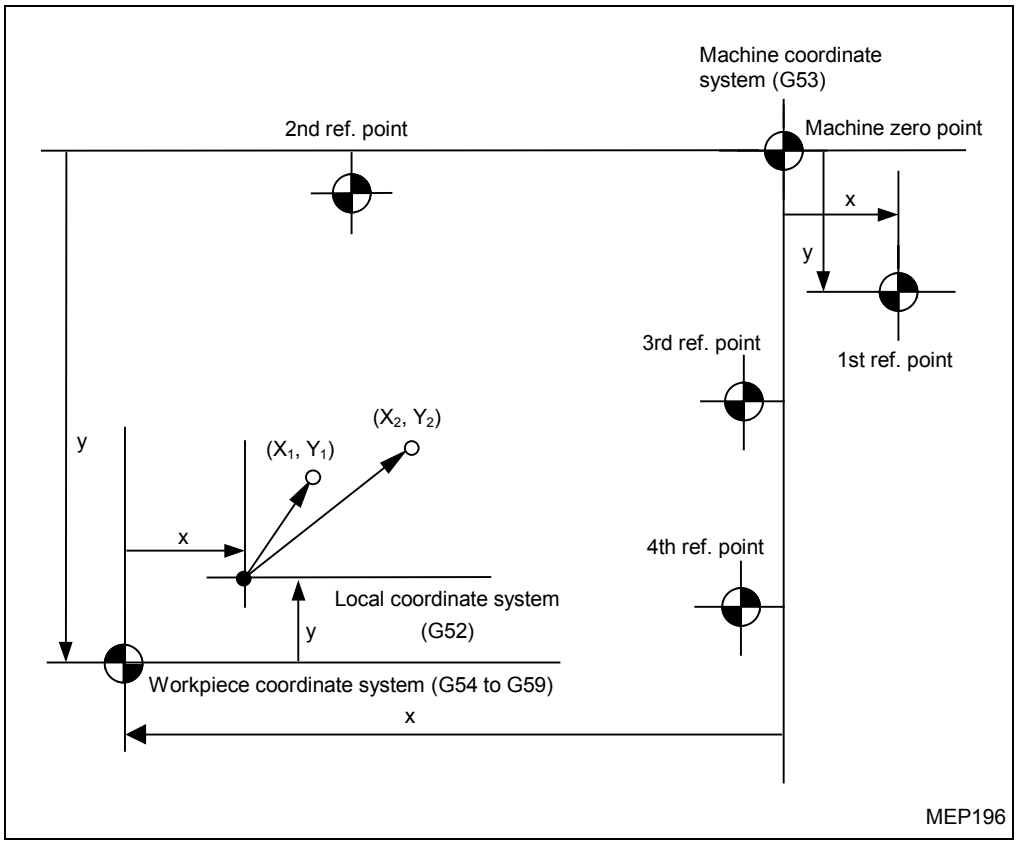


Fig. 2-52

2-12-3 Coordinate system setting: G92

1. Function and purpose

Setting of command G92 allows absolute coordinate system and current position display data to be newly preset exactly as programmed, without ever having to move the machine.

2. Programming format

```
G92 Xx Yy Zz;
```

3. Detailed description

- After power-on, return to the reference point is initially performed using watchdogs. Coordinate systems are set automatically on completion of return.

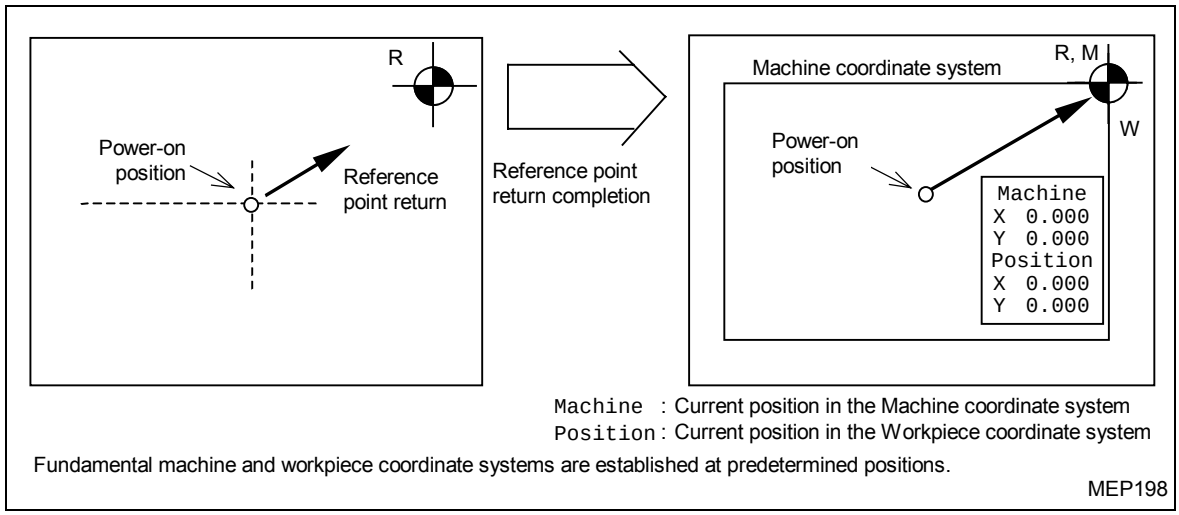


Fig. 2-53

- Setting of command G92 allows absolute (workpiece) coordinate system and current position display data to be newly preset exactly as programmed, without ever having to move the machine.

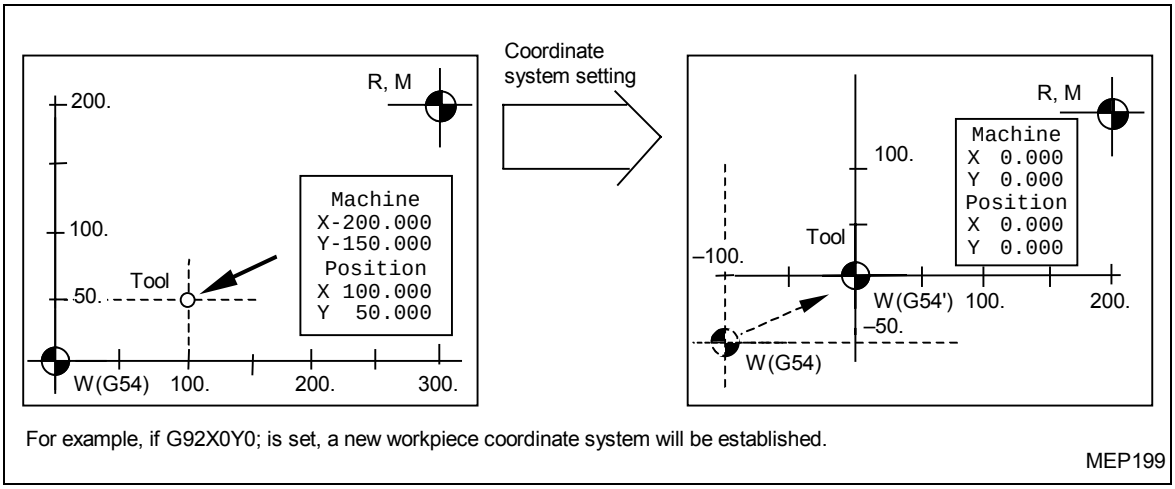


Fig. 2-54

2-12-4 Reference point return: G28

1. Function and purpose

G28 command first performs a G00 (rapid) positioning to the specified intermediate point along the specified axes, and then a returning to the first reference point at the rapid traverse rate independently along each specified axis.

2. Programming format

```
G28 Xx Yy Zz;
```

Use the addresses X, Y and Z to specify the coordinates of the intermediate point as required.

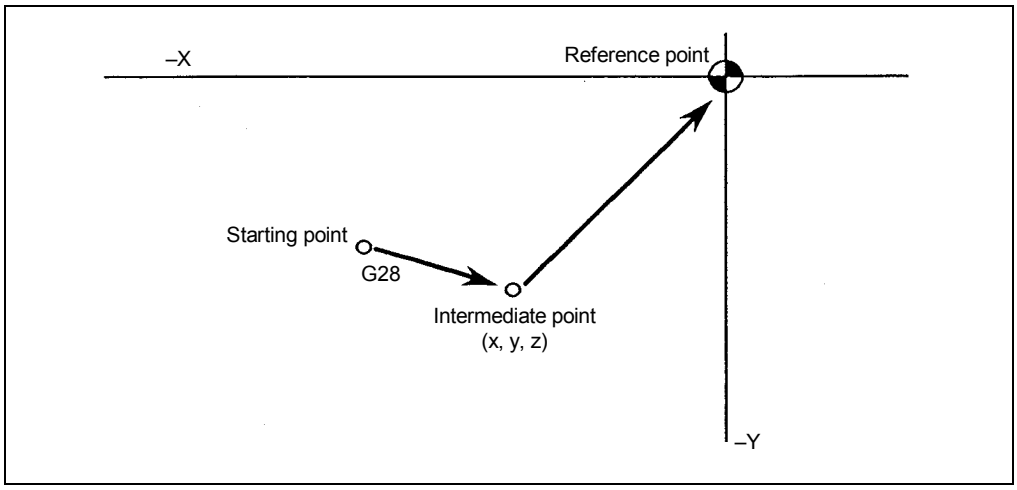


Fig. 2-55

2-12-5 Return to the 2., 3. or 4. reference point: G30

1. Function and purpose

The returning to the second, third, or fourth reference point can be programmed by setting “G30 P2 (P3, P4)”.

2. Programming format

```
G30 P2 (P3, P4) Xx Yy Zz;
```

Use the addresses X, Y and Z to specify the coordinates of the intermediate point as required.

3. Detailed description

1. Use address P to specify the number of the required reference point (P2, P3 or P4). Return to the second reference point is automatically selected if P-command is omitted.
2. Return to the second, third or fourth reference point is performed through the specified intermediate point like the return to the first reference point.

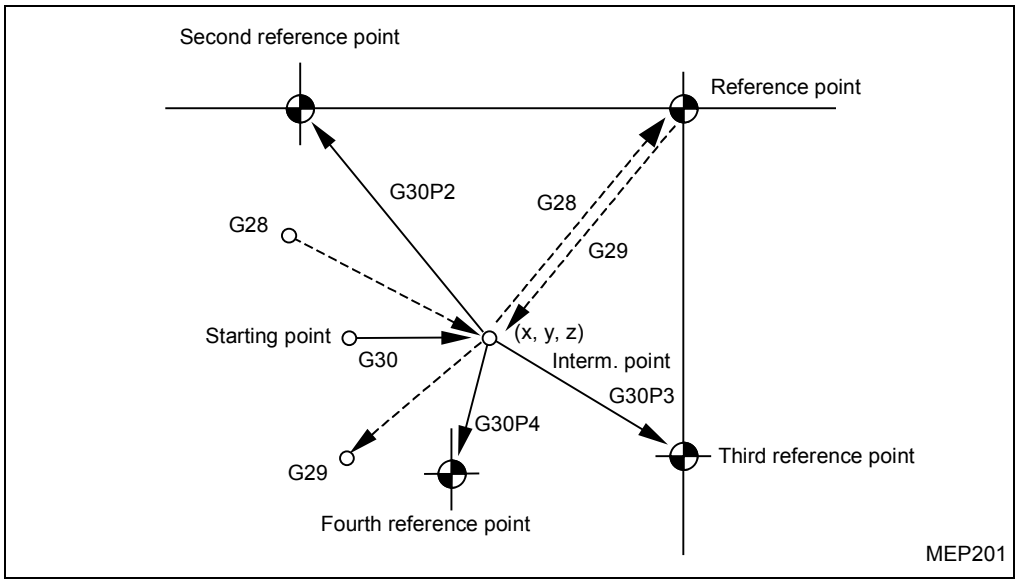


Fig. 2-56

2-12-6 Workpiece coordinate system selection: G54 to G59

1. Function and purpose

1. The workpiece coordinate system has its origin set to a specific point on the workpiece to be machined, and is used to simplify the machining program.
2. The absolute programming of the ending point has to refer to the current workpiece coordinate system (one of the systems G54 to G59).

2. Programming format

```
(G90) G54;
```

3. Detailed description

1. No selection of workpiece coordinate system is automatically done upon switching-on.
2. The functions G54 to G59 are modal ones.
3. The positions of the workpiece origins must have been set beforehand on the **WORK OFFSET** screen with respect to the machine zero point.

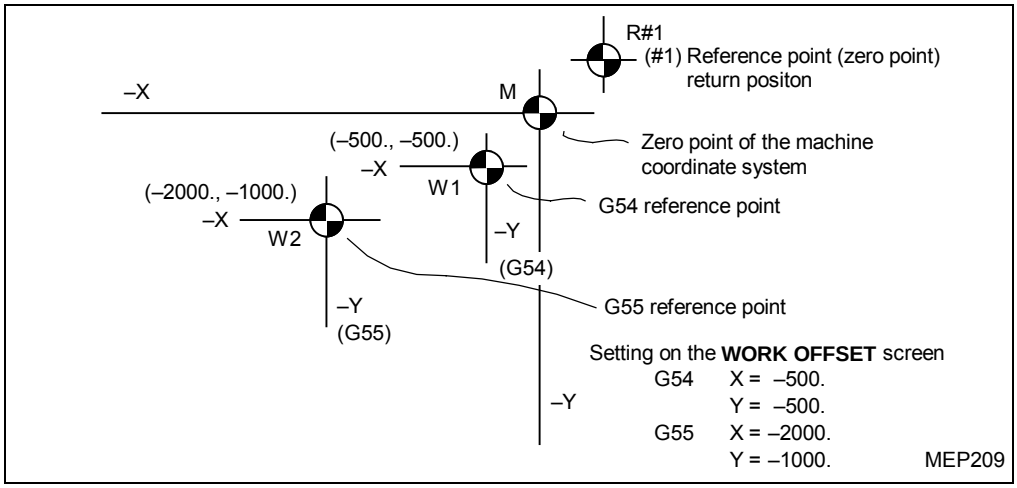


Fig. 2-57

4. The command of G92 does not affect the settings for the systems G54 to G59 on the **WORK OFFSET** screen. A motion command in the mode of G54, for example, is surely executed on the basis of the relevant original settings even if preceded by a shifting command with G92.

2-12-7 Local coordinate system setting: G52

1. Function and purpose

A local coordinate system in which the designated position becomes the program origin can be set on the current workpiece coordinate system by setting command code G52.
Command code G52 can also be used instead of G92 to update any positional shift between the machining program origin and the workpiece origin.

2. Programming format

```
G52 Xx Yy Zz;
```

3. Detailed description

Command G52, which causes no machine movement, is valid until a new G52 is issued. Command G52 is therefore convenient for using a new coordinate system without changing the origin of the workpiece coordinate system.

The local coordinate system offset data is cleared by execution of either an automatic reference-point return operation, or a manual reference-point (zero point) return operation using the watchdog method following power-on, as well as by a G92 command.

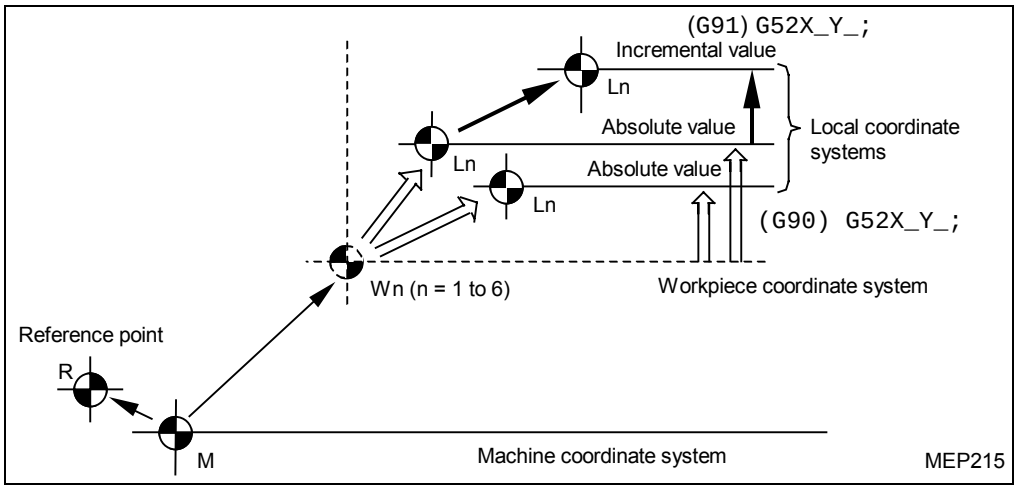


Fig. 2-58

Coordinates given in the absolute data input mode (G90) must refer to the local coordinate system.

Example 1: Setting local coordinates in the absolute data input mode
(Local coordinate system offset data is not integrated.)

```
[1] G28 X0 Y0;          [6] G00 X0 Y0;
[2] G00 G90 X1.Y1.;     [7] G01 X500 F100;
[3] G92 X0 Y0;          [8] Y500;
[4] G00 X500 Y500;      [9] G52 X0 Y0;
[5] G52 X1.Y1.;         [10] G00 X0 Y0;
```

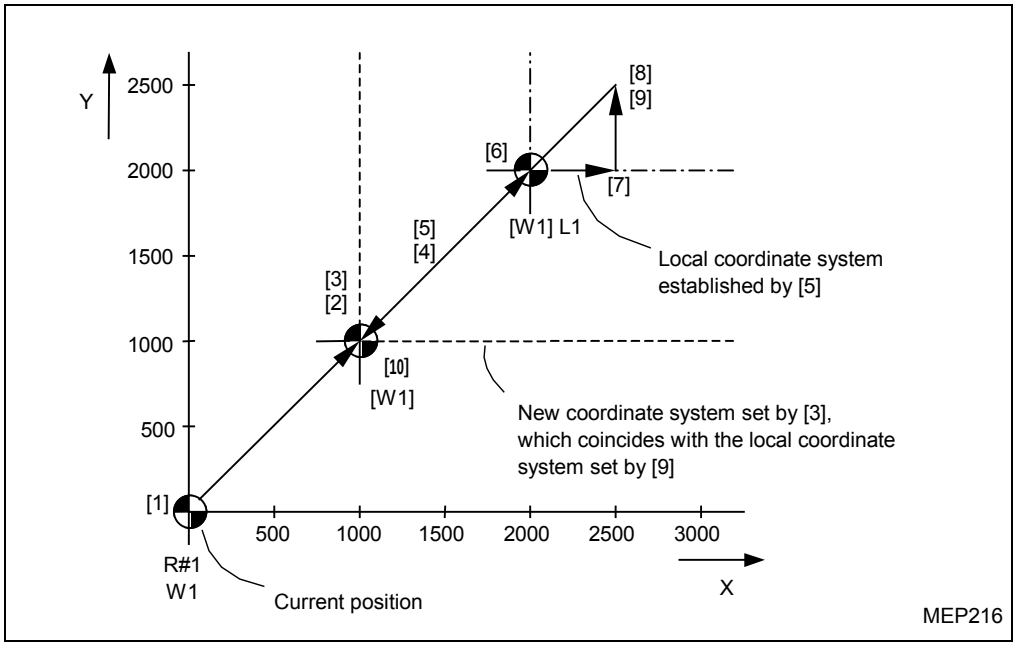


Fig. 2-59

The local coordinate system is established by [5] and cancelled by [9], and coincides with the coordinate system of [3].

Example 2: Setting local coordinates in the incremental data input mode
(Local coordinate system offset data is integrated)

```
[1]  G28X0Y0;              (a)  O100;
[2]  G92X0Y0;              (b)  G90G00X0Y0;
[3]  G91G52X500.Y500.;     (c)  G01X500.;
[4]  M98P100;              (d)  Y500.;
[5]  G52X1.Y1.;            (e)  G91;
[6]  M98P100;              (f)  M99;
[7]  G52X–1.5Y–1.5;
[8]  G00G90X0Y0;
      ⋮
     M02;
```

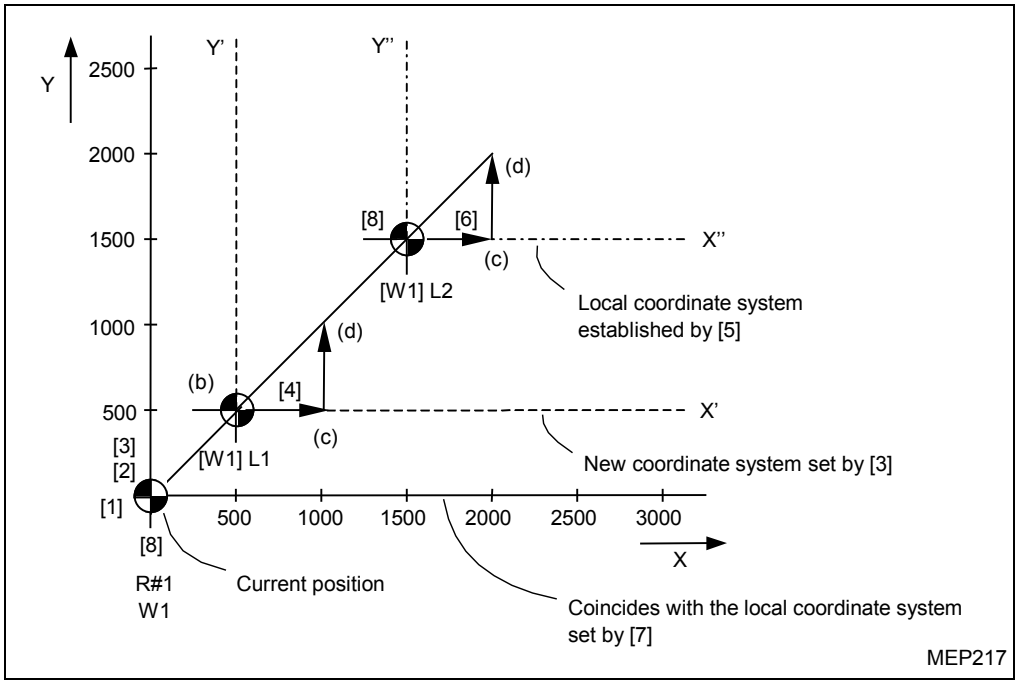


Fig. 2-60

Block [3] generates local coordinate system X'-Y' at the position of (500, 500) in the X-Y coordinate system.
Block [5] generates local coordinate system X"-Y" at the position of (1000, 1000) in the X'-Y' coordinate system.
Block [7] generates a local coordinate system at the position (–1500, –1500) in the X"-Y" coordinate system. That is, the local coordinate system and the X-Y coordinate system coincide, which means that the former has been cancelled.

Example 3: Combined use of local coordinate system and workpiece coordinate system

```
[1]  G28X0Y0;            (a)  O200;
[2]  G00G90G54X0Y0;      (b)  G00X0Y0;
[3]  G52X500.Y500.;      (c)  G01X500.F100;
[4]  M98P200;            (d)  Y500.;
[5]  G00G90G55X0Y0;      (e)  M99;
[6]  M98P200;                 %
[7]  G00G90G54X0Y0;
       ⋮
     M02;
```

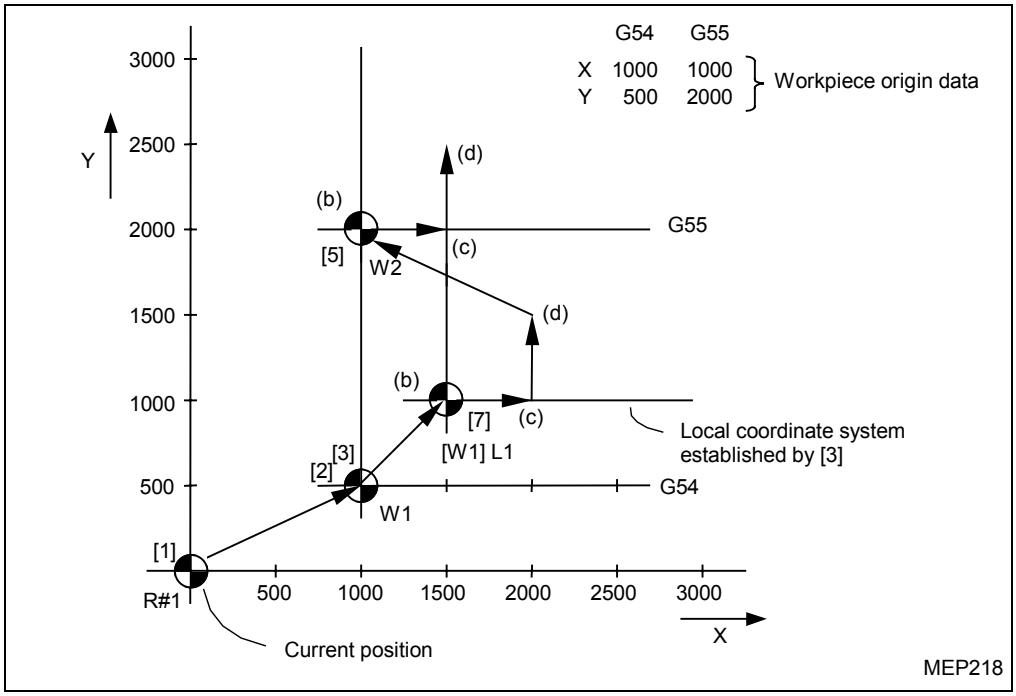


Fig. 2-61

Block [3] generates a local coordinate system at the position of (500, 500) in the G54 coordinate system no local coordinate system is generated in the G55 coordinate system.

Block [7] performs a movement to the local coordinate system reference point (zero point) of G54.

The local coordinate system can be cancelled using this format:

G90 G52 X0 Y0;

2-12-8 Machine coordinate system selection: G53

1. Function and purpose

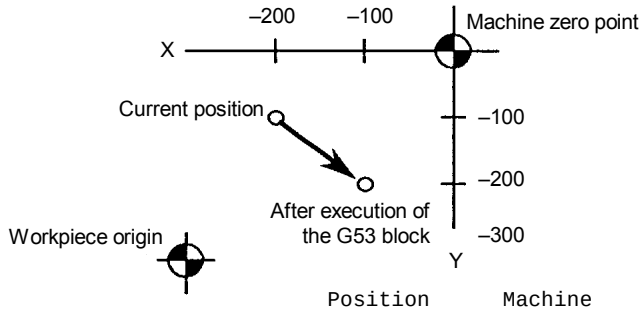
Use the G53 command, which is executed independently of the currently active workpiece coordinate system (set by G92, for example), to perform a positioning motion to a point which is easier to be described in the machine coordinate system.

2. Programming format

```
G53 Xx Yy Zz;
```

3. Detailed description

The function G53, which is non-modal, is provided for a rapid positioning to the target point that is to be specified in machine coordinates (absolute data input) in the same block with G53.




```
                                Position      Machine
G90;                              X     Y      X     Y
G92 X100. Y200.;        → 100.   200.  –200. -100.
G53 X–100.Y–200.; → 200.   100.  –100. -200.
```

Fig. 2-62

2-13 Laser Control Functions

2-13-1 Programmed cutting conditions: G82

1. Function and purpose

Command G82 is used to change the output intensity of the laser. Both piercing and cutting conditions can be changed with this command.

2. Programming format

```
G82 P_ Q_ R_ F_;
```

P: Laser peak output (1 - 1999 W)
Q: Pulse frequency (1 - 9999 Hz)
R: Pulse duty ratio (1 - 100%)
F: Rate of feed

3. Detailed description

1. Piercing and cutting condition commands can be specified in the same programming format. Address F can be set in a separate block.
2. Addresses P, Q, and R must be specified in the first G82 command in the program. They can be omitted for subsequent G82 commands. A program error occurs if addresses P, Q and R are not specified in the first G82 command.
3. The peak output values that can be specified with command G82 vary according to the type of machine. Refer to the specifications of your machine for details.
4. Laser emission does not occur at the time when command G82 is given. The laser is emitted with the conditions specified by command G82 after the discharge is turned on and the shutter opened.

4. Programming example

```
N1  G90 G92 X0 Y0;
N2  G82 P500 Q250 R25;          .....Piercing condition setting
N3  M51;                        .....Oxygen as assisting gas at low pressure
N4  M32;                        .....Z-axis positioning, tracing control ON
N5  M22;                        .....Discharge ON
N6  M20;                        .....Shutter open
N7  G04 X0.5;                   .....Piercing dwell
N8  G82 P500 Q300 R100 F3000;   .....Cutting condition setting
N9  G61 G01 X50.;
N10 Y50.;
N11 X0;
N12 Y0;
N13 M21;                        .....Shutter close
N14 M23;                        .....Discharge OFF
N15 M31;                        .....Tracing control OFF
N16 G91 G00 Z10.;               .....Z-axis return
N17 M30;
```

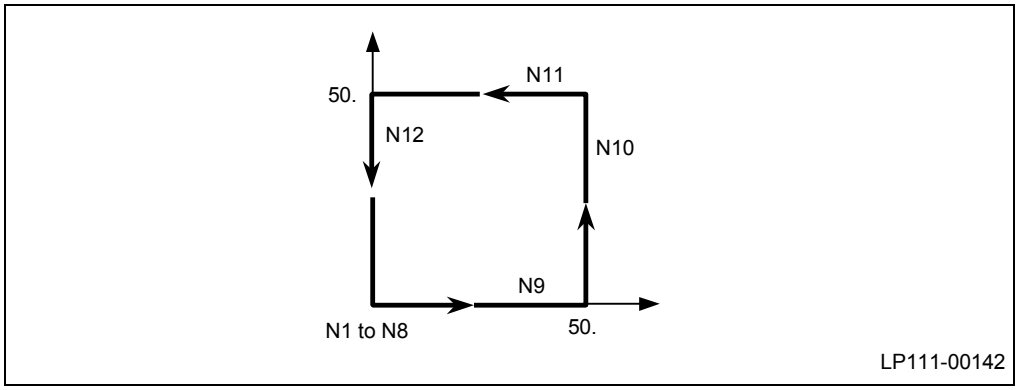


Fig. 2-63

2-13-2 Focal point control: G87

1. Function and purpose

Command G87 is used to change the focal point of the laser beam according to the material of workpieces and the thickness of plates.

2. Programming format

```
G87 P_;
```

P: Offset value of the focal point: (–99 to +99) × 0.1 [mm]
The setting range varies, depending on the tracer type.

3. Detailed description

1. When command G87 is executed, the gap length (N-W distance) specified on adjustment of the tracer is changed, and it causes the change of the focal point on the workpiece (see diagram below).
2. The setting range of offset varies, depending on the tracer type. The value beyond this range causes an alarm.

3. Pay attention to the focal point because it may be shifted by the gap control key.

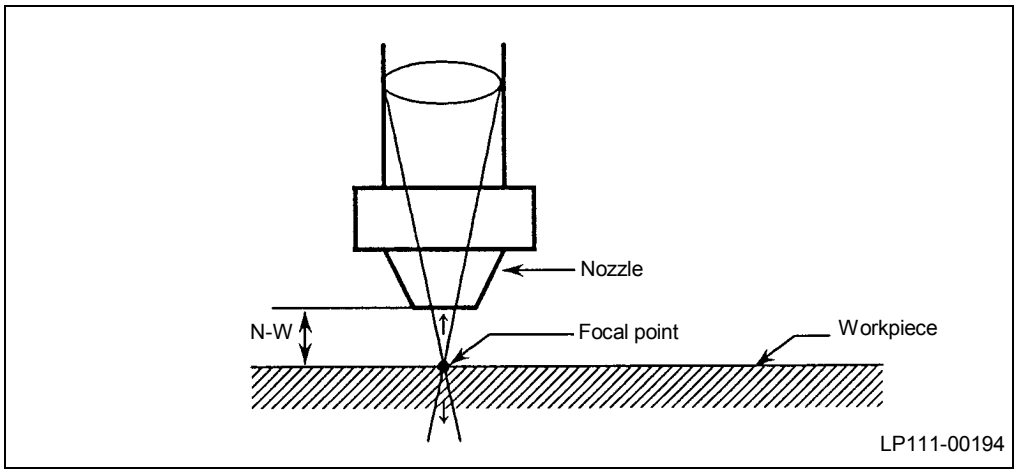


Fig. 2-64

4. Related parameters

Note: The parameter in question is an improvement introduced in the CNC, version v59.09.

Use the following parameter to select whether or not a block of G87P_ is to be completed without, or after, checking the change in the focal point.

User parameter No.	Function	Conditions of effectiveness	Setting
N113	Completion of a G87P_ block	Immediately	0 : Without checking the change 1 : After checking the change

Set **N113** to 0 to allow the next block to be executed without the change in the focal point being checked for the completion of a G87P_ block.
Set **N113** to 1 to prevent the next block from being executed before the change in the focal point has been checked for the completion of a G87P_ block.

2-13-3 Assisting-gas pressure control: G88

1. Function and purpose

Command G88 (an optional function) is used to specify and change the assisting-gas pressure.

2. Programming format

```
G88 P_;
```

P: Assisting-gas pressure (1 to 200) × 0.1 kgf/cm^2

Note 1: P: Assisting-gas pressure
Standard pressure: 1 to 99
Optional high-pressure electropneumatic specification: 1 to 200

Note 2: When an optional high-pressure electropneumatic regulator is provided, M54G88P (0 to 30) makes pressure unstable. Use M58G88P (0 to 30) to specify pressure data.

Note 3: The assisting-gas pressure achievable by setting M56 (M57 or M58) G88P (100 to 200) is up to 9.9 kgf/cm^2.

3. Detailed description

To set the cutting conditions in a program, the assisting gas is ordinarily specified by command G88 after commanding the M-code used to select the type of assisting gas. The setting value of gas pressure varies, depending on the machine specification.

4. Programming example

(When O_2 is used)

```
N1  G90 G92 X0 Y0;
N2  G82 P500 Q250 R25;
N3  M56;            .....Selection of oxygen as assisting gas
N4  G88 P05;        .....Assisting-gas pressure setting (0.5 kgf/cm2)
N5  M32;
N6  M20;
N7  M22;
N8  G04 X0.5;
N9  G88 P10;        .....Assisting-gas pressure change (1.0 kgf/cm2)
```

(When 4th assisting gas is used)

```
N1  G90 G92 X0 Y0;
N2  G82 P500 Q250 R25;
N3  M64;            .....4th assisting gas selection
N4  M54;            .....4th (3rd) assisting gas, high pressure selection
N5  G88P150;        .....Assisting-gas pressure setting (15 kgf/cm2)
```

Only for optional high-pressure electropneumatic specification

```
N6  M32;
N7  M20;
N8  M22;
N9  G04 X0.5;
N10 M58;            .....4th (3rd) assisting gas, low pressure selection
N11 G88 P10;        .....Assisting-gas pressure change (1.0 kgf/cm2)
```

2-14 Laser Control Programming Example

When programming, it is recommended to program the machining shape in a main program and the laser conditions, machining starting program and machining ending program in subprograms. Call the subprograms from the main program as required. This makes the whole programming easy. Programming method and values are different, depending on the workpiece shape, material and plate thickness.

Main program

```
O3000;
G90  G92 X2620.Y1260.; ..... Workpiece coordinate system setting
G52  X25.Y25.;         ..... Local coordinate system setting (program zero-point setting)
G00  X40.Y25.;         ..... Positioning in the XY-plane
M98  P3001;            ..... To call the subprogram for machining start
G61;                   ..... Exact stop mode (cutting mode setting)
G41  D4;               ..... Beam radius compensation
G01  X50.Y25.F3000;    }
Y50.;                  }
X0.;                   } Linear interpolation
Y0.;                   }
X50.;                  }
Y25.;                  }
M98  P3002;            ..... To call the subprogram for machining end
G28  X0.Y0.;           ..... Zero-point return
M30;                   ..... Program Reset & Rewind
```

Machining starting program

```
O3001;
M20;                   ..... Laser shutter open
G82  P1000 Q250 R20;   ..... Piercing condition setting
M68;                   ..... Super pulse selection
M56;                   ..... Selection of oxygen as assisting gas
G88  P10;              ..... Assisting-gas pressure setting
M32;                   ..... Tracing control ON
G87  P-5;              ..... Focal point change (lowered by 0.5 mm)
M22;                   ..... Discharge ON
G04  X0.1;             ..... Piercing dwell
G82  P600 Q250 R25;    ..... Cutting condition setting
M69;                   ..... Normal pulse selection
M56;                   ..... Selection of oxygen as assisting gas
G88  P15;              ..... Assisting-gas pressure change
G04  X0.5;             ..... 0.5 sec. waiting
M99;                   ..... Return to main program
```

Machining ending program

```
O3002;
M21;                    ..... Laser shutter close
M23;                    ..... Discharge OFF
M31;                    ..... Tracing control OFF
M09;                    ..... Assisting gas OFF
G91  G00 Z20.;          ..... Return Z-axis by incremental command
G90;                    ..... Return to absolute data input mode
G40;                    ..... Cancellation of beam radius compensation
M99;                    ..... Return to main program
```

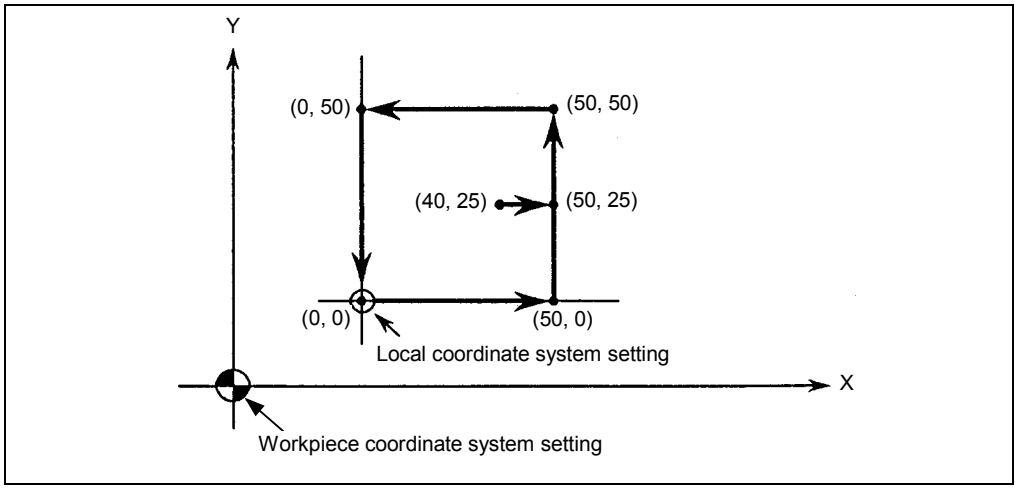


Fig. 2-65

2-15 Sample Program for High-Speed Cutting

High-speed cutting of an iron plate that does not require high machining accuracy and has a thickness of about 1 mm can be accomplished by creating a program as follows:

```
M20;                        .....Shutter open
G88  P36 M56;               .....Oxygen gas output
M32;                        .....Tracing ON
G82  P860 Q500 R100;        .....Laser output
M22;                        .....Discharge ON (see Note)
G02  I-5. F8000;            .....ϕ10 machining
G82  R0;
G00  X_Y_;
G82  R100;                  .....Duty 100%
G02  I-5.;
G82  R0;                    .....Duty 0%
G00  X_Y_;
    :
    :
```

Note: Instead of repetitive ON-OFF control for discharge by M22 and M23, the argument R in a block of G82 is to be changed as required under the condition of M22 (R100 for discharge ON, and R0 for discharge OFF) for a reduction in machining time.

The above program applies only for a sheet metal plate that does not require a piercing time.

Use M-codes if a piercing time is to be set.

- NOTE -

3 POWER RAMPING FUNCTION

3-1 Introduction

Power ramping is most effective during the cutting of an acute angle corner without lowering the cutting speed used in the cutting of straight sections.

3-1-1 Detailed description

To achieve high-speed cutting, a high rate of feed is required, usually with the maximum laser power. If cutting would be carried out at a fixed rate of feed, satisfactory high-speed cutting is possible. However, in actual cutting operation, axis motion speed is accelerated up to the designated rate of feed and then is decelerated where the direction of motion changes.

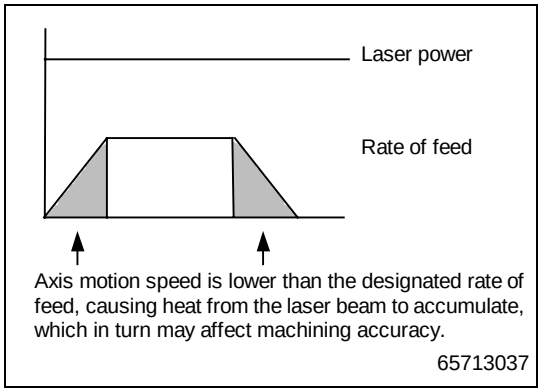

Laser power
Rate of feed
Axis motion speed is lower than the designated rate of feed, causing heat from the laser beam to accumulate, which in turn may affect machining accuracy.
65713037

Fig. 3-1 Rate of feed and laser power

Due to axis motion characteristics mentioned above, heat from the laser beam cannot be sufficiently dissipated at axis feed deceleration or acceleration areas and thus may affect cutting.

When cutting steel sheets or other materials with high conductivity, the heat generated in cutting a section can affect the cutting of succeeding sections.

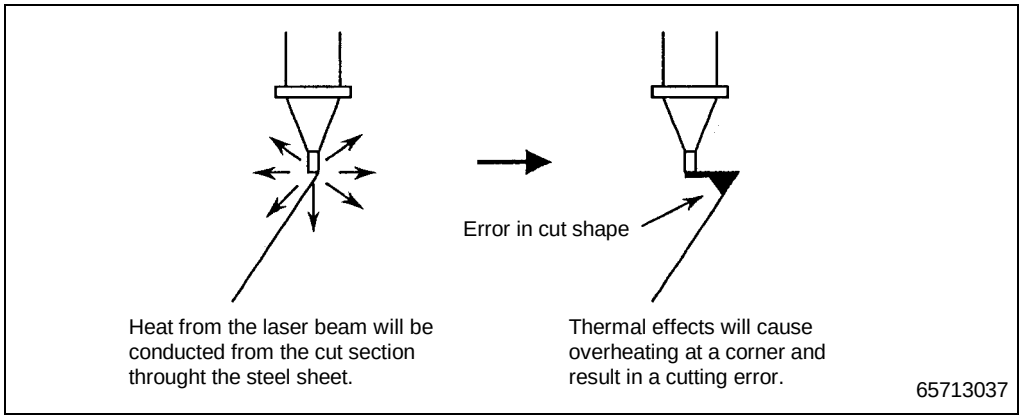

Error in cut shape
Heat from the laser beam will be conducted from the cut section throught the steel sheet.
Thermal effects will cause overheating at a corner and result in a cutting error.
65713037

Fig. 3-2 Cut section and thermal effects

To prevent such thermally induced problems, the axis motion speed is lowered and laser power output is reduced by means of pulse oscillation. Although this reduces the thermal effects during corner cutting, the total cutting speed is reduced accordingly resulting is low productivity. In addition, if such compensating operations are programmed, the program will require a lot of human labor and the program itself will be considerably long.

The power ramping function allows the laser oscillation mode to be changed to the pulse mode during the cutting of acute angle corners by simply adding several lines to the program, thus preventing heat accumulation and assuring high shape accuracy.

3-2 Programming

For this NC unit MAZATROL PREVIEW, G-codes used to call and cancel the power ramping function are G81 and G83.

G81 P (power) Q (frequency) R (duty) F (rate of feed) ; Power ramping ON

G83; Power ramping OFF

3-2-1 Program

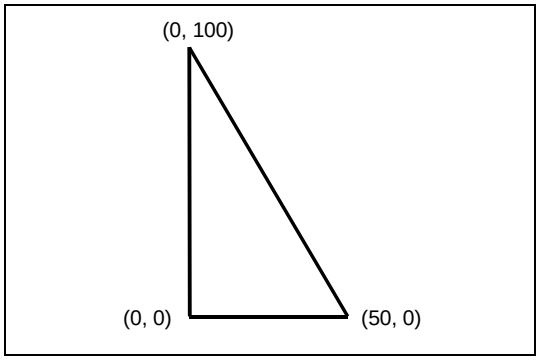


Fig. 3-3

Normal cutting example:

```
G92 X0. Y0.;
G00 X-5. Y0.;
M20;
G61;
G82 P500 Q250 R25; ..... Piercing
G88 P20 M56;
M32;
M22;
G04 X1.5;
G82 P20 M56;
G82 P500 Q250 R100 F3000; ..... Actual cutting
G01 X0.;
X50.;
X0. Y100.;
Y-5.;
M21;
M31;
M09;
G91 G0 Z10.;
G90 G0 X0. Y0.;
M30;
```

To cut entirely in the adaptive control ON mode:

```
G92 X0. Y0.;
G00 X-5. Y0.;
M20;
G61;
G82 P500 Q250 R25;  .....Piercing
G88 P20 M56;
M32;
M22;
G04 X1.5;
G88 P20 M56;
G82 P500 Q250 R100 F3000;
G81 P500 Q250 R25 F300;
G01 X0.;
X50.;
X0. Y100.;
Y-5.;
M21;
M31;
M09;
G83;                .....Adaptive control OFF
G91 G0 Z10.;
G90 G0 X0. Y0.;
M30;
```

Note: Data for addresses F, P and R in the G81 block must satisfy the following conditions.

G81F ≤ G82F Rate of feed
P ≤ P Laser peak power
R ≤ R Pulse duty

G81 mode must be cancelled for the block of piercing conditions.

3-3 Setting Conditions

How to determine the cutting conditions:

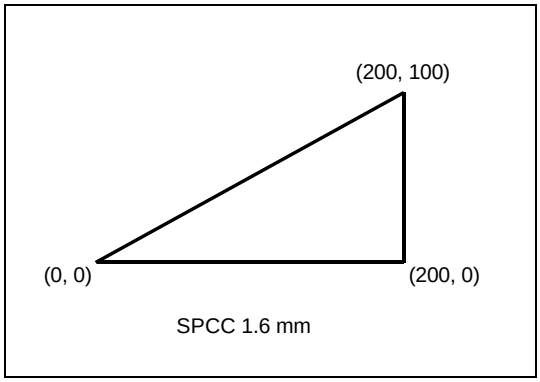


Fig. 3-4

When cutting a workpiece as illustrated above:

1. Set the laser power to the maximum rate and select the cutting conditions that give the best cutting state for straight lines at the highest rate of feed.
2. Select the conditions in which the same shape can be cut at F300 - F500 in pulses and the corner can be acutely angled without dulling even when the assist gas pressure is the same as for high-speed cutting.
3. Set the conditions obtained in step (1) as the G82 block conditions and those in step (2) as the G81 block conditions.

The assist gas pressure is to be set beforehand by selecting and M-code in the MDI mode, or by a G88-command.

3-4 Parameter

The SETTING parameter contains the following item.

ADAPT. CONTROL LENGTH 10.00 mm

This indicates the deceleration zone distance. Change the setting within the range from 1 to 99.0 mm as required.

3-5 Real-time Power Ramping Function

The real-time power ramping function changes the laser duty according to the particular speed of the nozzle end during acceleration/deceleration.
Real-time power ramping is accomplished by changing the duty in terms of designated speed - real speed ratio (See Fig. 3-5.).

This function is valid only for X-axial/Y-axial cutting feed mode. This function does not operate for Z-axis operation (copy machining) or piercing-related laser commands.

`G181 R__;` Power ramping ON
`G183;` Power ramping cancel

R: Minimum override duty (0% to 100%)

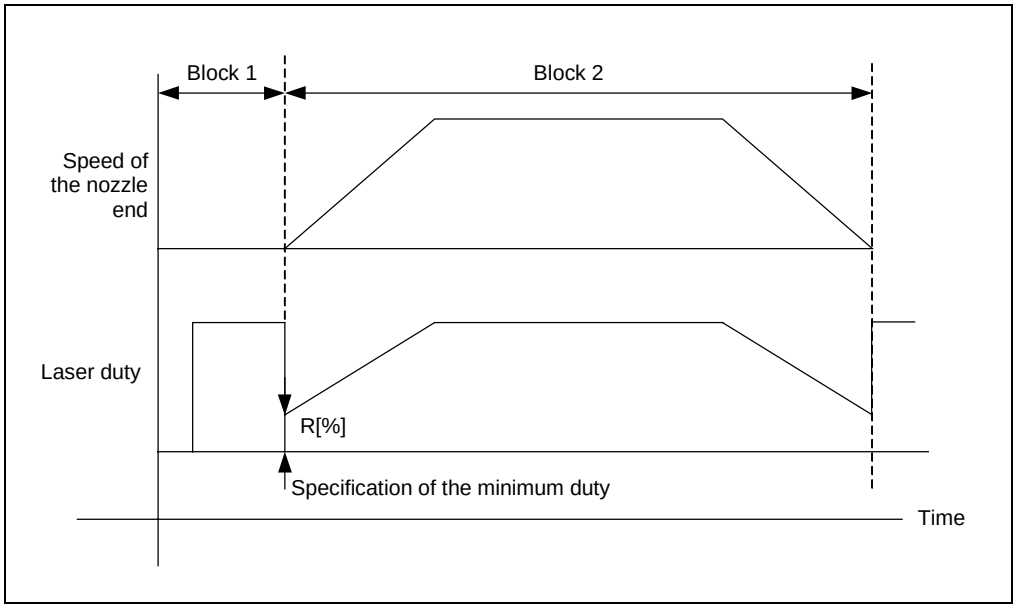


Fig. 3-5 Axial speed and laser output

3-5-1 Sample programs

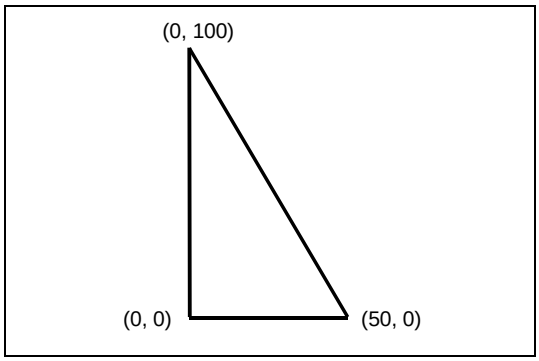


Fig. 3-6

Normal cutting example:

```
G92 X0. Y0.;
G00 X-5. Y0.;
M20;
G61;
G82 P500 Q250 R25; ..................Piercing
G88 P20 M56;
M32;
M22;
G04 X1.5;
G82 P20 M56;
G82 P500 Q250 R100 F3000; .....Actual cutting
G01 X0.;
X50.;
X0. Y100.;
Y-5.;
M21;
M31;
M09;
G91 G0 Z10.;
G90 G0 X0. Y0.;
M30;
```

To cut entirely in the adaptive control ON mode:

```
G92 X0. Y0.;
G00 X-5. Y0.;
M20;
G61;
G181 R5; ..............................Real-time power ramping ON
G82 P500 Q250 R25;.....Piercing
G88 P20 M56;
M32;
M22;
G04 X1.5;
G88 P20 M56;
G82 P500 Q250 R100 F3000;
G01 X0.;
X50.;
X0. Y100.;
Y-5.;
M21;
M31;
M09;
G183; ..............................Real-time power ramping OFF
G91 G0 Z10.;
G90 G0 X0. Y0.;
M30;
```

3-5-2 Related parameters

The following parameters allow the selection of the initial mode, and the specification of the minimum duty, for real-time power ramping:

User parameter No.	Function	Conditions of effectiveness	Setting
N003	Modal selected by power ON and NC reset	When power is turned on	0 : G183 1 : G181
N004	Minimum override duty	Immediately	0 to 100

Note 1: If 0 is assigned as the minimum override duty, cutting may fail since no laser beam will be emitted at the beginning and end of cutting.

Note 2: Parameter **N004** will be used if argument R is not included in the G181 command.

- NOTE -

4 MACHINING MODE SELECTION (for the HG and HTX)

4-1 Overview

The enhancement of the NC unit's processing capacity allows high-accuracy machining to be achieved at a higher speed. The three available machining modes are: standard machining mode for machining with the ordinary accuracy at a higher speed, high-speed machining mode for machining at the highest possible speed, and high-accuracy machining mode for machining with higher accuracy at the ordinary speed.

Note 1: The function of machining mode selection is an improvement introduced in the CNC, version v59.09.

Note 2: High-speed machining mode is initially selected as a default value for the CNC, from version v59.09 onward.

Note 3: This function is only valid for the mode of sheet metal working (G22J0 to G22J2.3).

4-2 Programming Format

```
G108 P_;
```

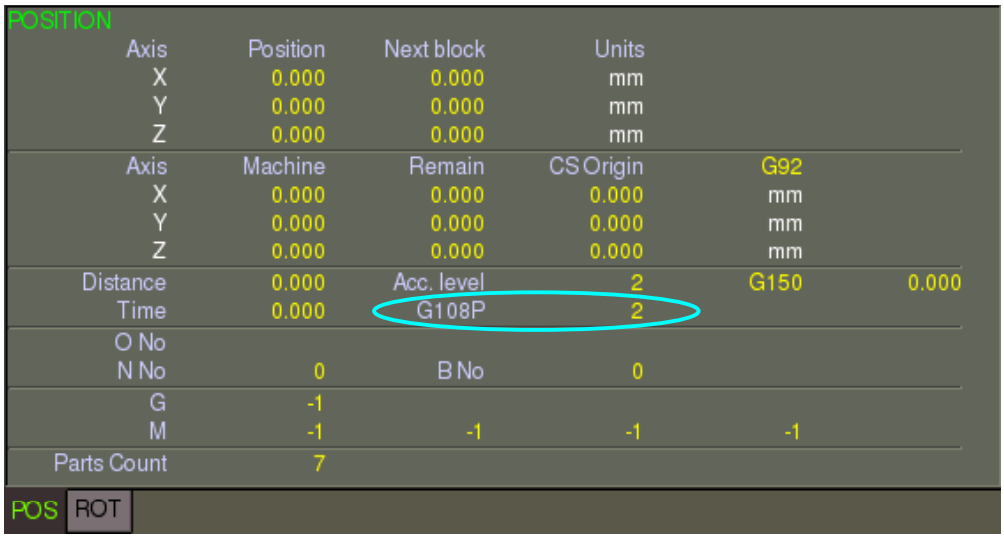
0: High-accuracy machining mode

1: Standard machining mode

2: High-speed machining mode

4-3 Detailed Description

The currently active machining mode is displayed on the **POSITION** page of the **COMMAND** screen, as shown below:


POSITION
Axis Position Next block Units
X 0.000 0.000 mm
Y 0.000 0.000 mm
Z 0.000 0.000 mm
Axis Machine Remain CS Origin G92
X 0.000 0.000 0.000 mm
Y 0.000 0.000 0.000 mm
Z 0.000 0.000 0.000 mm
Distance 0.000 Acc. level 2 G150 0.000
Time 0.000 G108P 2
O No
N No 0 B No 0
G -1
M -1 -1 -1 -1
Parts Count 7
POS ROT

The **G108P** values 0, 1, and 2 indicate the machining mode selection as follows:

0: High-accuracy machining mode
Machining can be done more accurately in the same period of machining time as before.

1: Standard machining mode
The time of machining with the same accuracy as before can be significantly shortened.

2: High-speed machining mode (Default setting)
The machining time can be shortened as far as the machine's efficiency allows.

4-4 Related Parameters

Use the following parameter to select the initial machining mode (to be automatically selected by resetting). The default setting is two (2).

User parameter No.	Function	Conditions of effectiveness	Setting
N080	Initial machining mode (to be selected by resetting)	Immediately	0 : High-accuracy machining mode 1 : Standard machining mode 2 : High-speed machining mode

Set **N080** to 0, 1, and 2, respectively, in order that High-accuracy machining mode, Standard machining mode, and High-speed machining mode may be automatically selected at the end of a machining program as well as by pressing the Reset key.

5 ELLIPTICAL POSITIONING

5-1 Overview

In the conventional method of laser machining, rapid transition from one machining shape to the next one is performed (to prevent collision between torch and workpiece) in the following operating sequence:

Tracing OFF → Raising the torch on the Z-axis → Rapid positioning in the XY-plane → Tracing ON → Lowering the torch on the Z-axis.

The new function of elliptical positioning serves the purpose of shortening the intermediate positioning time and thus contributes toward reducing the total machining time.

The function of elliptical positioning allows any block of rapid positioning in the XY-plane, in the mode of tracing control, to be executed with simultaneous Z-axis up-and-down movements automatically added and, as a result, reduces the number of command blocks required.

Note: This function is an improvement introduced in the CNC, version v59.10.

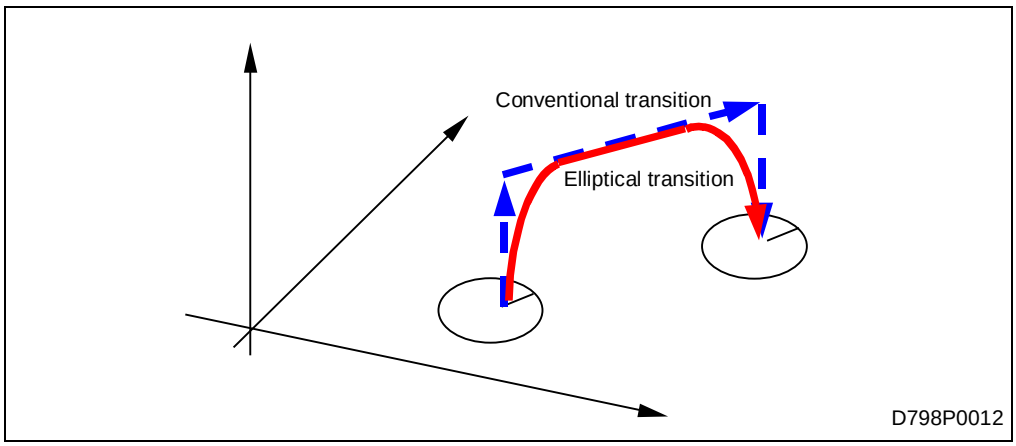


Fig. 5-1

5-2 Programming Format

`G153 Z_;` Elliptical positioning ON

`G150;` Elliptical positioning OFF

Specify the Z-axis raising distance with address Z in the G153 block.

5-3 Operation

Since the elliptical positioning always occurs in the mode of tracing control, positioning is safely and correctly done to the required height even above a bent or uneven workpiece.

1. When the workpiece is bent

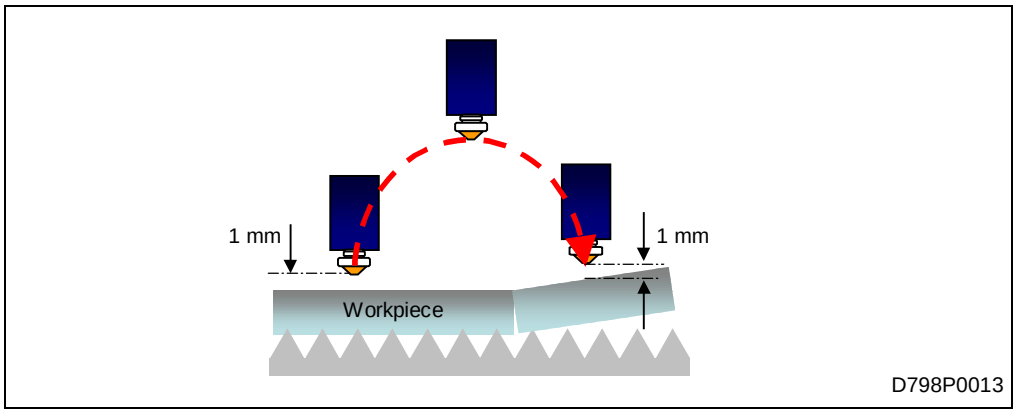


Fig. 5-2

2. When the workpiece is uneven

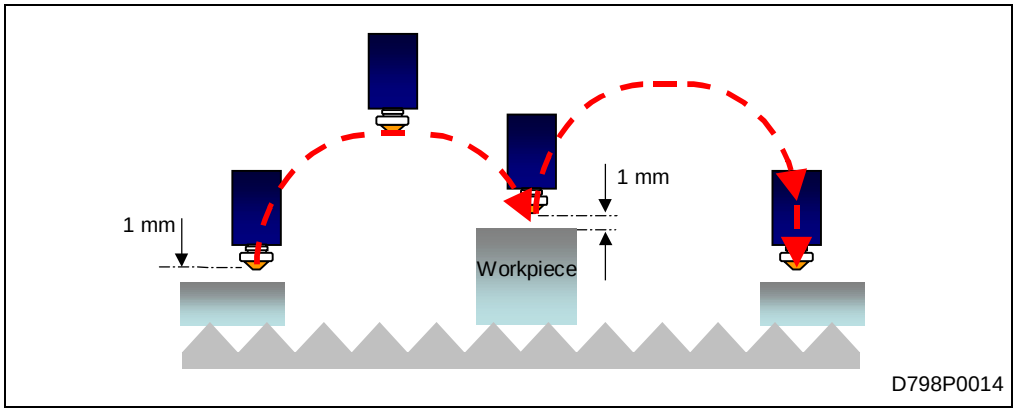


Fig. 5-3

The collision with the workpiece cannot be avoided in the following cases:

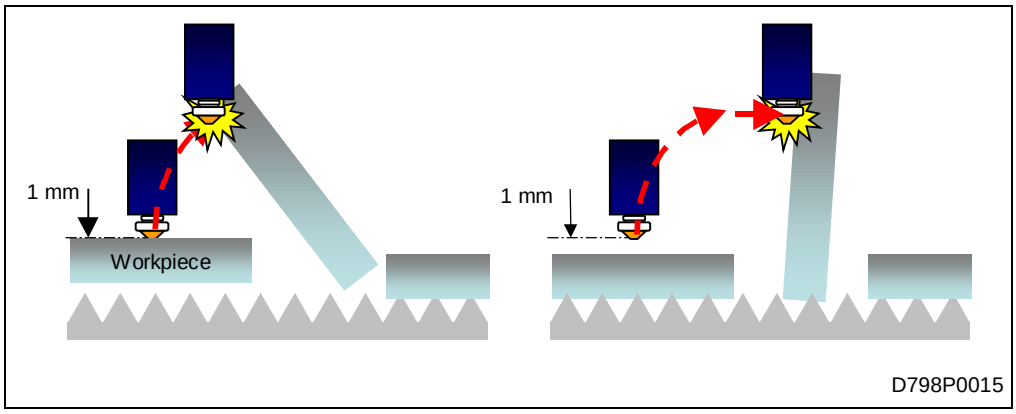
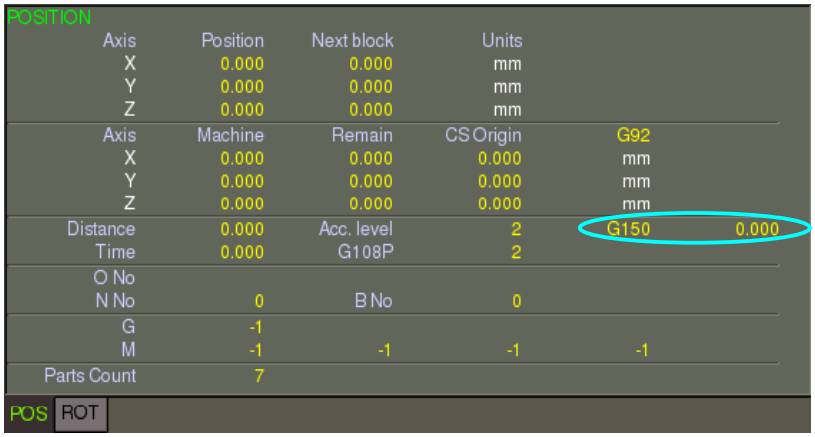


Fig. 5-4

5-4 Detailed Description

1. The mode of G153 is retained until it is cancelled by G150. The initial modal setting is G150 (automatically selected by resetting).
2. Argument Z can be omitted in the G153 block if the setting of parameter **N085** is to be used.
3. The value of argument Z is retained until it is overwritten by another command of G153Z_.
4. The elliptical positioning only takes place at blocks of G00 (rapid positioning) in the mode of M32 (Tracing ON). It does not apply to the other movement blocks (with G01, G02, G03, G28, G30, G31, or G53).
5. The modal information about elliptical positioning is displayed on the **POSITION** page of the **COMMAND** screen. **G153Z** is displayed with the specified raising distance in the mode of elliptical positioning.


POSITION
Axis Position Next block Units
X 0.000 0.000 mm
Y 0.000 0.000 mm
Z 0.000 0.000 mm
Axis Machine Remain CS Origin G92
X 0.000 0.000 0.000 mm
Y 0.000 0.000 0.000 mm
Z 0.000 0.000 0.000 mm
Distance 0.000 Acc. level 2 G150 0.000
Time 0.000 G108P 2
O No
N No 0 B No 0
G -1
M -1 -1 -1 -1
Parts Count 7
POS ROT

6. A command of G153Z0 produces no simultaneous movements on the Z-axis.

Note: The alarms relevant to the elliptical positioning are as follows:

- Any (explicit) Z-axis motion command in the mode of M32 (Tracing ON) will cause the alarm **176 ILLEGAL Z COMMAND**. Cancel the tracing function beforehand to give a Z-axis motion command.
- Entering a negative value with address Z in a G153 block will cause the alarm **601 ILLEGAL NUMBER INPUT**.
- In case the specified raising distance should result in the Z-axis software motion limit being overstepped, the operation will be stopped with the alarm **+SOFT OT** before the execution of the motion block in question.

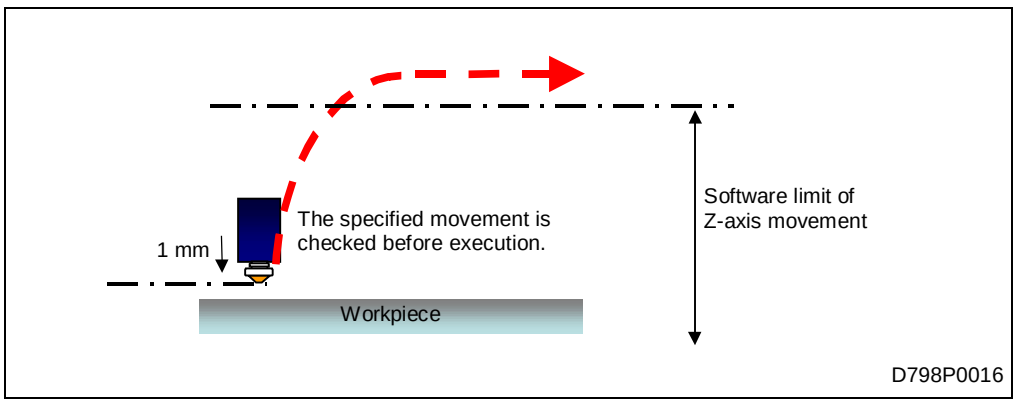


Fig. 5-5

5-5 Related Parameters

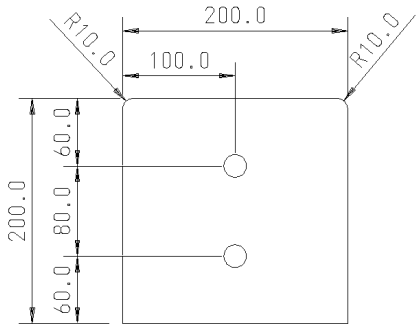
Use the following parameter to set the initial raising distance (to be automatically set by re-setting) for the elliptical positioning. The default setting is zero (0).

User parameter No.	Function	Conditions of effectiveness	Default setting
N085	Initial Z-axis raising distance (to be set by resetting)	Immediately	0

5-6 Programming Example

The example below shows a case where the elliptical positioning is only used in positioning for the contour cutting (no elliptical positioning for hole-machining sections).

```
O0001
G22I0.J2.3
G90G92X0.Y0.Z0.
G64
M20
G82P0Q500R0
M22
N1
G153Z15. .....Elliptical positioning ON.
                Raising distance is set to 15 (mm).
G90G00X100.7Y131.98 .....No elliptical positioning since tracing
                control is not ON.
G87P10
M32.....Tracing ON.
G82P1300Q150R20
G04X1.5
G181R100
G88P20
G82P1000Q1000R100F3500
G87P0
G04X0.5
G42D08G01X100.808Y130.033
G02I-0.808J9.967
G01G40
G82P0Q500R0
G88P06
G87P20
N2
G153Z0. .....Raising distance changed to 0 (mm).
G90G00X106.868Y55.801.....No elliptical positioning.
G87P10
G82P1300Q150R20
G04X1.5
G88P20
G82P1000Q1000R100F3500
G87P0
G04X0.5
G42D08G01X108.499Y54.731
G02I-8.499J5.269
G01G40
G82P0Q500R0
G88P06
G87P20
```




```
N3
G153Z15. ..... Raising distance changed
                to 15 (mm).
G90G00X200.Y-1.95 ..... Elliptical positioning.
G87P10
G82P1300Q150R20
G04X1.5
G88P20
G82P1000Q1000R100F3500
G87P0
G04X0.5
G42D08G01Y-0.05
Y190.
G03X190.Y200.R10.
G01X10.
G03X0.Y190.R10.
G01Y0.
X200.05
G40
G82P0Q500R0
G88P06
G87P0
M23
M09
M21
M31..... Tracing OFF.
G150..... Elliptical positioning OFF.
G91G00Z50.
M30
```

	G153Z_		G150	
	Tracing ON	Tracing OFF	Tracing ON	Tracing OFF
G00X_Y_	⊙	○	○	○
G00Z_	×	○	×	○

⊙: Elliptical positioning ○: Normal rapid positioning ×: Operation stopped with alarm **176 ILLEGAL Z COMMAND**

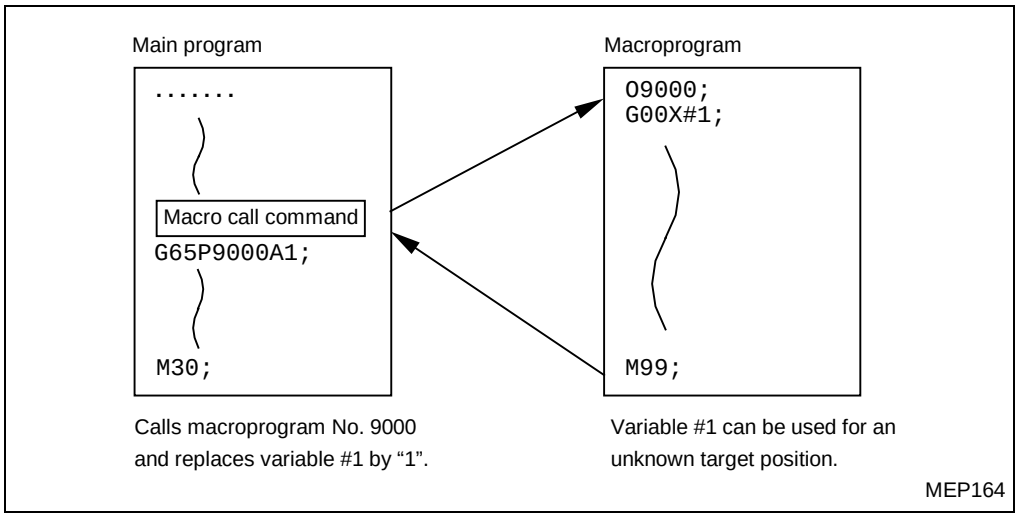
- NOTE -

6 USER MACROS

6-1 Introduction

Macroprogram call, data calculation, data input to/output from a personal computer, data control, judgment, branching, and various other instructions can be used with variables commands to perform measurements and other operations.

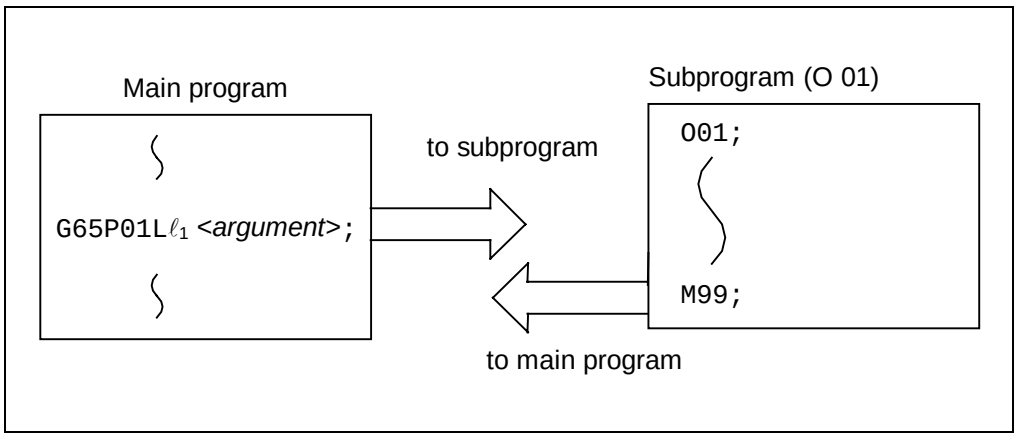
A macroprogram is a subprogram which is created using variables, calculation instructions, control instructions, etc. to have special control features.
These special control features (macroprograms) can be used by calling them from the main program as required. These calls use macro call instructions.



6-2 Macro Call Instructions

Two types of macro call instructions are provided: single-call instructions used to call only at the designated block, and modal call instructions used to call at each block within a macro call mode. Modal call instructions are further divided into type A and type B.

1. Single call



The designated user macro subprogram ends with M99.
Instruction G65 calls the designated user macro subprogram only once.

Format:
G65 P__L__<*argument*> ;

Repeat times

Program No.

<Argument>
When argument is to be delivered to the user macro subprogram as a local variable, designate the required data with the respective addresses.

A. Argument specification Ⅰ

Format: A_B_C_ ····· X_Y_Z_

Detailed description

- An argument can be specified using all addresses, except G, L, N, O, and P.
- Except for I, J, and K, addresses does not need be specified in an alphabetical order.
  I_J_K_ ... Correct
  J_I_K_ ... Wrong (See the description of “Argument specification Ⅱ”.)

Relationship between address and variables number		Relationship between address and variables number	
Address specified using method Ⅰ	Variable in macro-program	Address specified using method Ⅰ	Variable in macro-program
A	#1	Q	#17
B	#2	R	#18
C	#3	S	#19
D	#7	T	#20
E	#8	U	#21
F	#9	V	#22
H	#11	W	#23
I	#4	X	#24
J	#5	Y	#25
K	#6	Z	#26
M	#13		

B. Argument specification II

Format: A_B_C_I_J_K_I_J_K_·····

Detailed description

- Up to a maximum of 10 sets of arguments that each consist of addresses I, J, and K, as well as A, B, and C, can be specified.
- If identical addresses overlap, specify them in the required order.

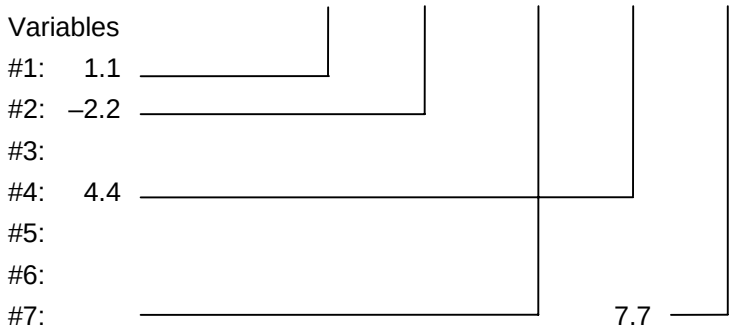
Argument specification II addresses	Variables in macro-programs	Argument specification II addresses	Variables in macro-programs
A	#1	J5	#17
B	#2	K5	#18
C	#3	I6	#19
I1	#4	J6	#20
J1	#5	K6	#21
K1	#6	I7	#22
I2	#7	J7	#23
J2	#8	K7	#24
K2	#9	I8	#25
I3	#10	J8	#26
J3	#11	K8	#27
K3	#12	I9	#28
I4	#13	J9	#29
J4	#14	K9	#30
K4	#15	I10	#31
I5	#16	J10	#32

Note: In the table above, the numerals 1 through 10 have been added to addresses I, J, and K just to denote the order of arrangement of the designated sets of arguments: these numerals are not included in actual instructions.

C. Combined use of argument specification I and II

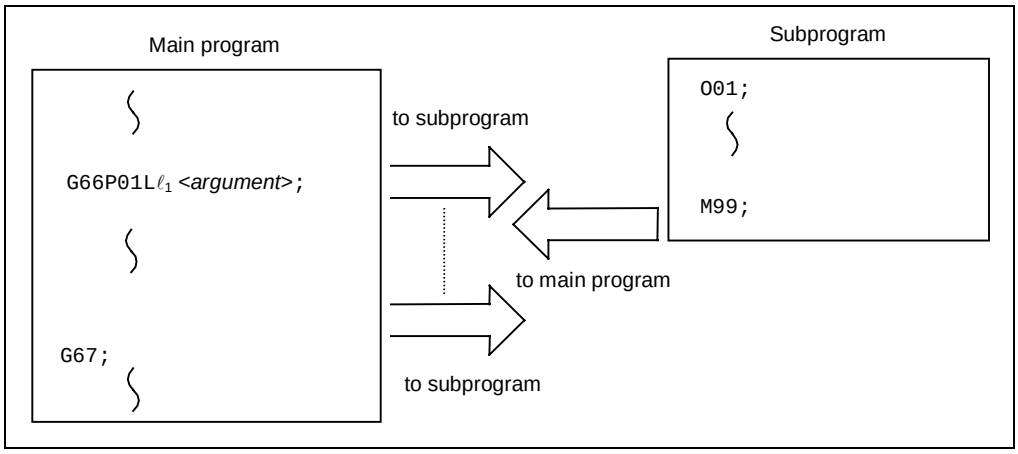
When both method I and method II are used to specify arguments, only the latter of two arguments which have an address corresponding to the same variable will become valid.

Example: Call command `G65 A1.1 B–2.2 D3.3 I4.4 I7.7`



If two arguments (D3.3 and I7.7) are designated for the variable of #7, only the latter argument (I7.7) will be used.

2. Modal call, type A (Move command call)



For a block that has a move command code between G66 and G67, the designated user macro subprogram is executed after that move command has been executed. Execution of the macro program is repeated as specified with L (ℓ_1) for every call.
For modal call of type A, the methods of specifying <argument> are the same as used for single call.

Format:
G66 P__L__<*argument*> ;

Repeat times

Program number

Detailed description

- When command G66 is entered, the designated user macro subprogram will be called every time after execution of the move commands within a block until command G67 (cancellation) is entered.
- Command code G67 can be left out.

3. Modal call, type B (Block-to-block call)

The designated user macro subprogram is called unconditionally for each of the command blocks present between G66.1 and G67. Execution of the macro program is repeated as specified with L for every call.

Format:
G66.1 P__L__<*argument*> ;

Repeat times

Program number

Detailed description

1. In the G66.1 mode, only the codes O, N, and G in each of the read command blocks are executed. Codes other than O, N, and G are handled as arguments.
2. All significant blocks in the G66.1 mode are regarded as preceded by the command G65P_. For example, the block of

   `N100G01G90X100. Y200. F400R1000`

   in the G66.1P1000 mode is handled as equivalent to

   `N100G65P1000G01G90X100. Y200. F400R1000.`

   Note: Call is executed even for a G66.1 command block in the G66.1 mode, with the relationship between the addresses of the arguments and the variables numbers being the same as for G65 (single call).
3. The data range of the G, L, P, and N commands that you can set as new variables using the G66.1 mode is the same as the data range of usual NC commands.
4. Sequence number N, modal G-codes, and O are all updated as modal information.

4. G-code macro call

The user macro subprograms of the required program number can be called just by setting G-codes.

Format:
`G××` *<argument>* `;`

G-code for macro call

Detailed description

1. The instruction shown above performs the same function as those of the instructions listed below. Which of these listed instructions will apply is determined by the parameter data to be set for each G-code.

   `M98PΔΔΔΔ;`

   `G65PΔΔΔΔ` *<argument>*`;`

   `G66PΔΔΔΔ` *<argument>*`;`

   `G66.1PΔΔΔΔ` *<argument>*`;`
2. Use parameters (User Parameter J) to set the relationship between G×× (macro call G-code) and PΔΔΔΔ (program number of the macro to be called).
3. Of G00 through G255, up to a maximum of 10 command codes can be used with this instruction unless the uses of these codes are clearly predefined by EIA Standards, such as G00, G01, G02, etc.
4. A macro call G-code cannot be included in user macro subprograms that have been called using G-codes.

5. Auxiliary command macro call (M- or T-code macro call)

The user macro subprograms of the required program number can be called just by setting M- or T-codes.

Format:
`Mm;` (or `Tt;`)

M (or T) code for macro call

Detailed description (The following description also applies to T-codes.)

1. The instruction shown above performs the same function as those of the instructions listed below. Which of these listed instructions will apply is determined by the parameter data to be set for each M-code.

   `M98PΔΔΔΔ;`

   `G65PΔΔΔΔMm;`

   `G66PΔΔΔΔMm;`

   `G66.1PΔΔΔΔMm;`

2. Use parameters to set the relationship between Mm (macro call M-code) and PΔΔΔΔ (program number of the macro to be called).
   Up to a maximum of 10 M-codes, ranging from M00 to M95, can be registered. Do not register the M-codes that are fundamentally required for your machine, nor M0, M1, M2, M30, and M96 through M99.
3. As for T-codes, only one code can be used for macro call.
4. As is the case with M98, M-code output and MF signal output are skipped for a macro call M-code (which is included in the program data display on the screen).
5. If registered auxiliary command codes are set in the user macro subprograms that have been called using M-codes, macro calls will not occur since those special auxiliary command codes will be handled as usual ones (M- or T-codes).

6. Differences in usage between commands M98, G65, etc.

- Arguments can be designated for G65, but cannot be designated for M98.
- Sequence numbers can be designated for M98, but cannot be designated for G65, G66, or G66.1.
- The level of local variables is fixed for M98, but for G65 does change according to the depth of nesting. (For example, #1s, if present before and after M98, always mean the same, but if present before and after G65, they have different meanings.)
- Command M98 can have up to a maximum of nine levels of call multiplexity when combined with G65, G66, or G66.1, whereas the maximum available number of levels for command G65 is four when it is combined with G66 or G66.1.

7. Multiplexity of macro call commands

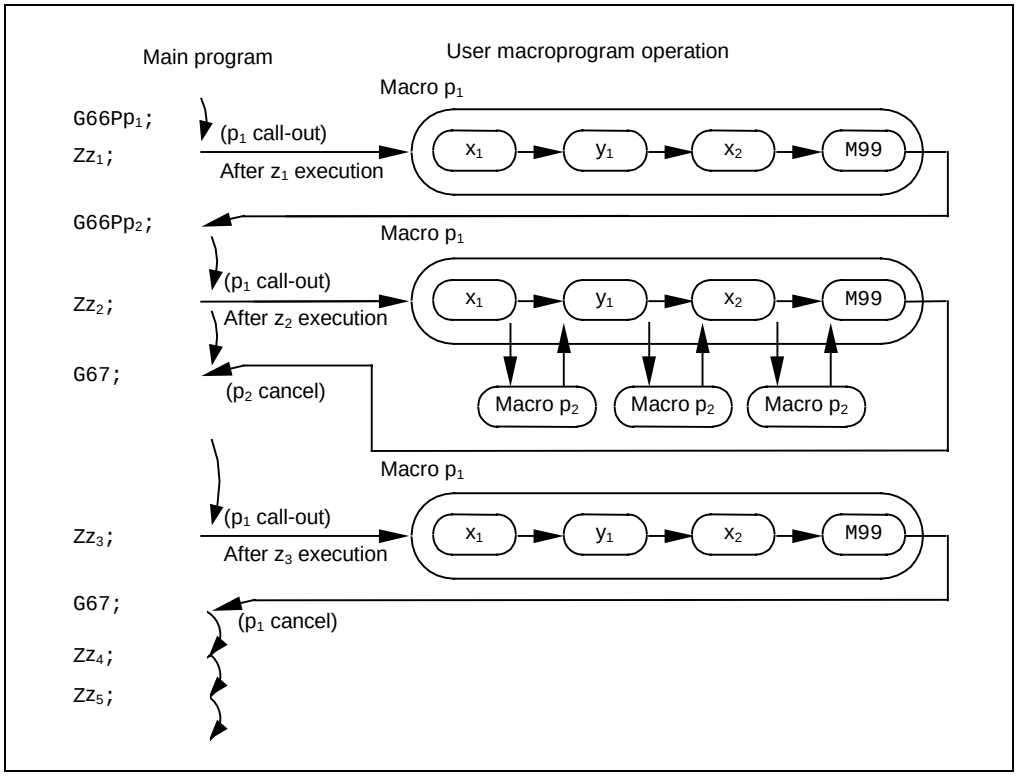
The maximum available number of levels of macro subprogram call is four, whether it is single or modal. Arguments in macro call instructions become valid only within the level of the called macro. Since the multiplexity of macro call is of up to a maximum of four levels, arguments can be included in a program as local variables each time a macro call is made.

Note 1: When a G65, G66, or G66.1 macro call or an auxiliary command macro call is made, nesting will be regarded as single-level and thus the level of local variables will also increase by 1.

Note 2: For modal call of type A, the designated user macro subprogram is called each time a move command is executed. If, however, multiple G66s are present, the next user macro subprogram will be called even for the move commands in the macro each time axis movement is done.

Note 3: User macro subprograms are cancelled in a reverse order to that in which they have been arranged.

Example:



6-3 Variables

6-3-1 Outline

Of all types of variables available for the NC unit, only local variables, common variables, and part of system variables are retained even after power-off.

1. Multiplexing of variables

Under user macro specifications, variables can have their identifiers (identification numbers) either transformed into variables, which is referred to as multiplexing, or replaced with <expression>.

Example 1: Multiplexing variables

```
#1=10 #10=20 #20=30;   From #1 = 10, #[#[#1]] = #[#10] will result.
#5=#[#[#1]];           From #10 = 20, #[#10] = #20 will result. Therefore #5 = #20, i.e. #5
                       = 30 will result.
#1=10 #10=20 #20=30;   From #1 = 10, #[#[#1]] = #[#10] will result.
#5=1000;               From #10 = 20, #[#10] = #20 will result. Therefore #20 = #5, i.e.
#[#[#1]]=#5;           #20 = 1000 will result.
```

Example 2: Replacing variables identifiers with <expression>

```
#10=5;
#[#10+1]=1000;         #6 = 1000 will result.
#[#10–1]=–1000;        #4 = –1000 will result.
#[#10*3]=100;          #15 = 100 will result.
#[#10/2]=100;          #2 = –100 will result.
```

2. Undefined variables

Under user macro specifications, variables remaining unused after power-on or local variables that are not argument-specified by G65, G66, or G66.1 can be used as <empty>. Also, variables can be forcibly made into <empty>.
Variable #0 is always used as <empty> one, and this variable cannot be defined on the left side of the expression.

A. Arithmetic expression

```
#1=#0; .....#1 = <empty>
#2=#0+1; .....#2 = 1
#3=1+#0; .....#3 = 1
#4=#0*10; .....#4 = 0
#5=#0+#0; .....#5 = 0
```

Note: Be careful that <empty> is handled the same as 0 during processing of expressions.
<empty> + <empty> = 0
<empty> + <constant> = constant
<constant> + <empty> = constant

B. Applying variables

Application of an undefined variable alone causes even the address to be ignored.

If #1 = <empty>

```
G0X#1Y1000;          is equivalent to G0Y1000, and
G0X[#1+10]Y1000;     is equivalent to G0X10Y1000.
```

Note: As is the case with the replacement of variables identifiers, expressions must be put in brackets for an application.

C. Conditional expression

Only for EQ and NE, does <empty> differ from 0 in meaning.

If #101 = <empty>		If #101 = 0	
#101EQ#0	<empty> = <empty> holds.	#101EQ#0	0 = <empty> does not hold.
#101NE0	<empty> ≠ 0 holds.	#101NE0	0 ≠ 0 does not hold.
#101GE#0	<empty> ≥ <empty> holds.	#101GE#0	0 ≥ <empty> holds.
#101GT0	<empty> > 0 does not hold.	#101GT0	0 > 0 does not hold.

Hold-conditions and not-hold-conditions list
(For conditional expressions including undefined variables)

Right / Left side	EQ		NE		GT		LT		GE		LE	
	Empty	Constant	Empty	Constant	Empty	Constant	Empty	Constant	Empty	Constant	Empty	Constant
Empty	H			H				H	H		H	
Constant			H		H				H			

H: Holds (The conditional expression holds.)
Blank: The conditional expression does not hold.

6-3-2 Types of variables

1. Common variables (#100 to #199, and #500 to #699)

Common variables refer to the variables to be used in common at any position. The identifiers of common variables which can be used are from #100 to #199, or from #500 to #699.

2. Local variables (#1 to #32)

Local variables refer to variables that can be defined as <argument> when calling a macro subprogram, or those which can be used locally within the main program or a subprogram. There is no relationship between macros. Thus, these variables can be overlapped on each other, but up to a maximum of four levels of overlapping.

G65Pp_1Lℓ_1 <*argument*> ;

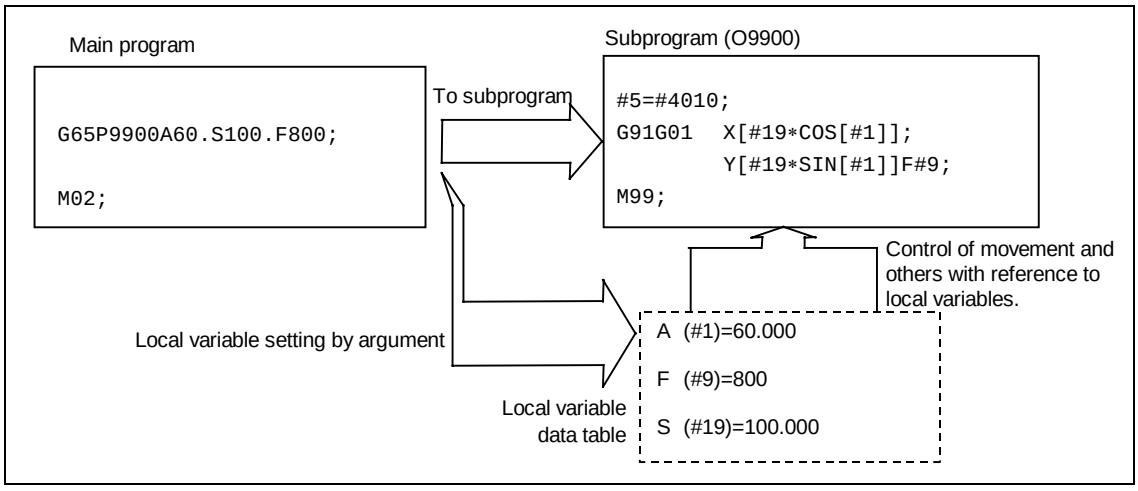
where p_1: Program number
ℓ_1: Number of repeat times
<Argument> must be: Aa_1 Bb_1 Cc_1 ··· Zz_1.

The following represents the relationship between the address specified by <argument> and the local variables number used in the user macro unit:

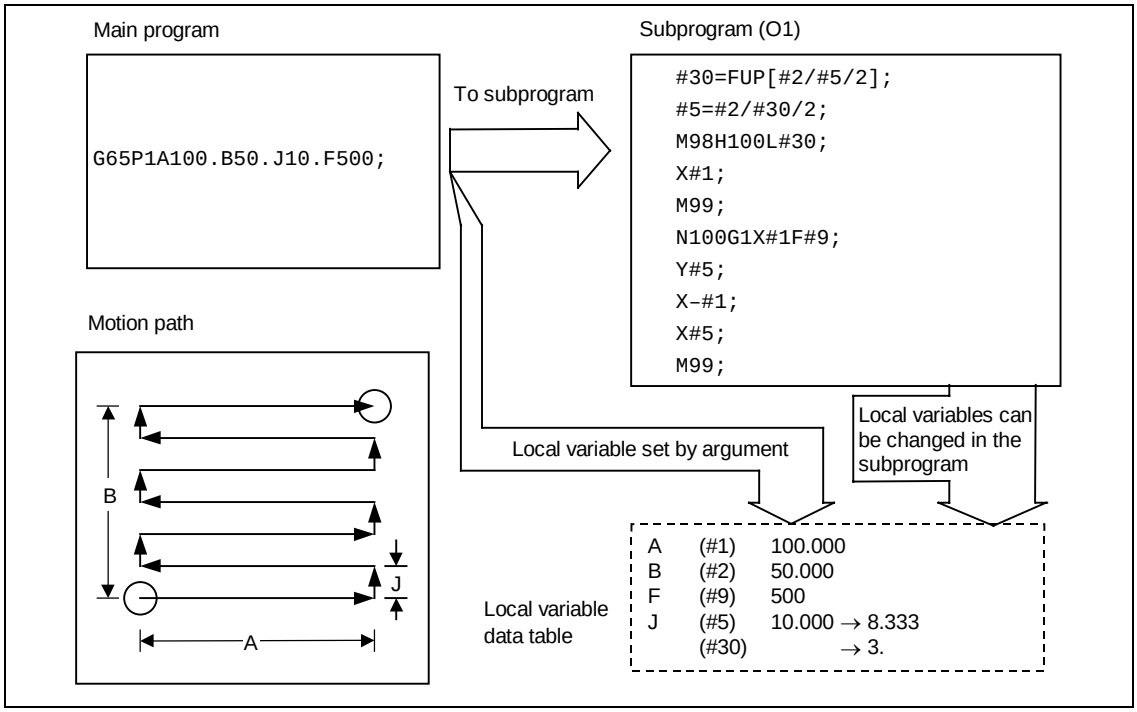
Call commands		Argument address	Local variable	Call commands		Argument address	Local variable
G65 G66	G66.1			G65 G66	G66.1		
O	O	A	#1	O	O	R	#18
O	O	B	#2	O	O	S	#19
O	O	C	#3	O	O	T	#20
O	O	D	#7	O	O	U	#21
O	O	E	#8	O	O	V	#22
O	O	F	#9	O	O	W	#23
×	×*	G	#10	O	O	X	#24
O	O	H	#11	O	O	Y	#25
O	O	I	#4	O	O	Z	#26
O	O	J	#5			–	#27
O	O	K	#6			–	#28
×	×*	L	#12			–	#29
O	O	M	#13			–	#30
×	×*	N	#14			–	#31
×	×	O	#15			–	#32
×	×*	P	#16				
O	O	Q	#17				

Argument addresses marked as × in the table above cannot be used. Only during the G66.1 mode, however, can argument addresses marked with an asterisk (*) in this table be additionally used. Also, the dash sign (–) indicates that no address is crosskeyed to the local variables number.

1. Local variables for a subprogram can be defined by specifying <argument> when calling a macro.

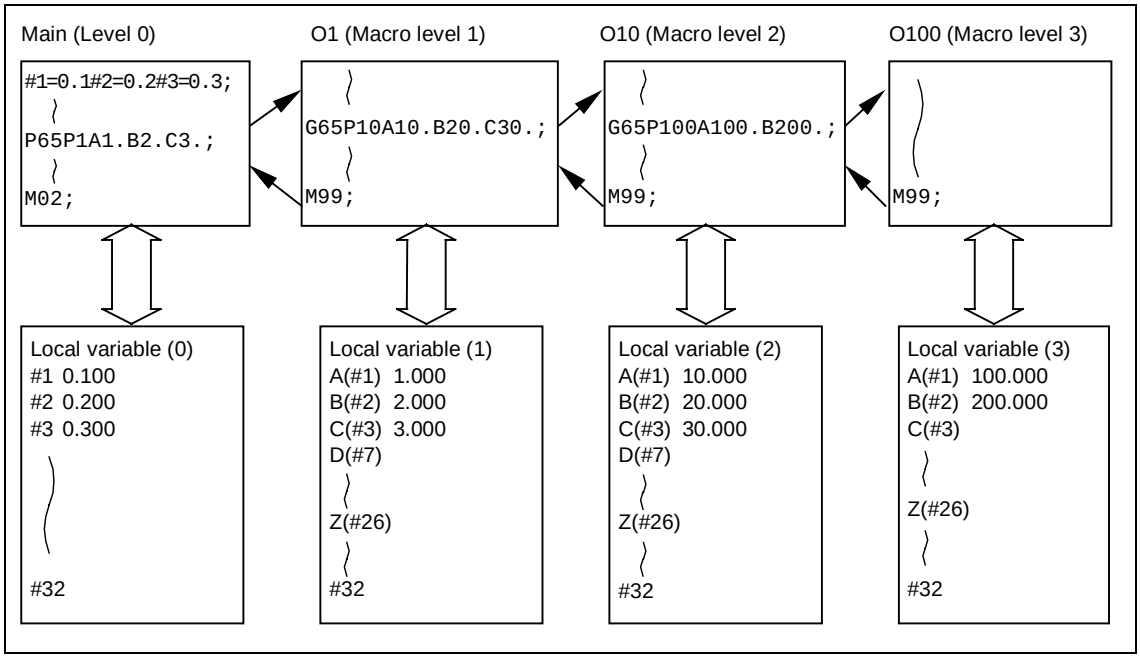


2. Within a subprogram, local variables can be freely used.



In the sample program shown above, although the argument J has initially been programmed as a machining pitch of 10 mm, it has been changed into 8.333 mm to ensure equal-pitched machining.
Also, local variable #30 contains the calculated data about the number of times of reciprocal movement.

3. Local variables can be used for each of the four levels of macro call separately. For the main program (macro level 0), separate local variables are also provided. The local variables of level 0, however, cannot be designated with arguments.



How the local variables are currently being used is displayed on the screen.
For further details, refer to the Operating Manual.

3. Macro interface input system variables (#1000 to #1035)

You can check the status of an interface input signal by reading the value of the appropriate variables number (#1000 to #1035).
The read value of the variables number is either 1 (contact closed) or 0 (contact open). You can also check the status of all input signals of the variables from #1000 to #1031 by reading the value of variables number 1032. Variables from #1000 to #1035 can only be read; they cannot be placed on the left side of an arithmetic expression.

System variable	Points	Interface input signal	System variable	Points	Interface input signal
#1000	1	O21.00	#1016	1	O21.16
#1001	1	O21.01	#1017	1	O21.17
#1002	1	O21.02	#1018	1	O21.18
#1003	1	O21.03	#1019	1	O21.19
#1004	1	O21.04	#1020	1	O21.20
#1005	1	O21.05	#1021	1	O21.21
#1006	1	O21.06	#1022	1	O21.22
#1007	1	O21.07	#1023	1	O21.23
#1008	1	O21.08	#1024	1	O21.24
#1009	1	O21.09	#1025	1	O21.25
#1010	1	O21.10	#1026	1	O21.26
#1011	1	O21.11	#1027	1	O21.27
#1012	1	O21.12	#1028	1	O21.28
#1013	1	O21.13	#1029	1	O21.29
#1014	1	O21.14	#1030	1	O21.30
#1015	1	O21.15	#1031	1	O21.31

System variable	Points	Interface input signal
#1032	32	O21
#1033	32	O22
#1034	32	O23
#1035	32	O24

4. Macro interface output system variables (#1100 to #1135)

You can send an interface output signal by assigning a value to the appropriate variables number (#1100 to #1135).
All output signals can take either 0 or 1.
You can also send all output signals of the variables from #1100 to #1131 at the same time by assigning a value to variables number 1132 (2^0 to 2^{31}). In addition to the data writing for offsetting the #1100 to #1135 output signals, the reading of the output signal status can be done.

System variable	Points	Interface output signal	System variable	Points	Interface output signal
#1100	1	I34.00	#1116	1	I34.16
#1101	1	I34.01	#1117	1	I34.17
#1102	1	I34.02	#1118	1	I34.18
#1103	1	I34.03	#1119	1	I34.19
#1104	1	I34.04	#1120	1	I34.20
#1105	1	I34.05	#1121	1	I34.21
#1106	1	I34.06	#1122	1	I34.22
#1107	1	I34.07	#1123	1	I34.23
#1108	1	I34.08	#1124	1	I34.24
#1109	1	I34.09	#1125	1	I34.25
#1110	1	I34.10	#1126	1	I34.26
#1111	1	I34.11	#1127	1	I34.27
#1112	1	I34.12	#1128	1	I34.28
#1113	1	I34.13	#1129	1	I34.29
#1114	1	I34.14	#1130	1	I34.30
#1115	1	I34.15	#1131	1	I34.31

System variable	Points	Interface output signal
#1132	32	I34
#1133	32	I35
#1134	32	I36
#1135	32	I37

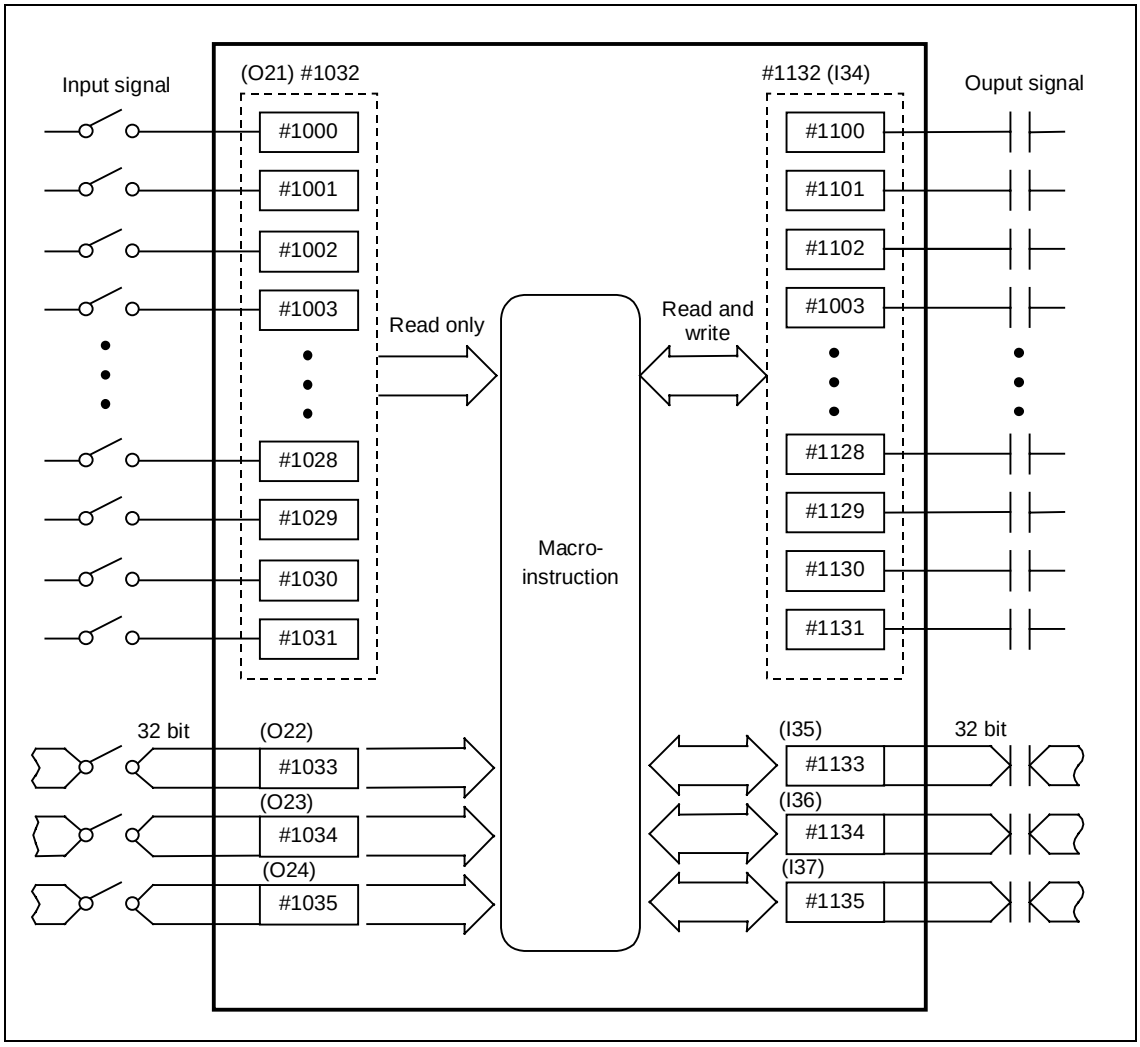
Note 1: Data of the system variables from #1100 to #1135 is saved according to the logical level (1 or 0) of the signal that has been lastly sent. The saved data is cleared by power-on/off automatically.

Note 2: The following applies if a data other than 1 or 0 is assigned to the variables from #1100 to #1131:

<empty> is regarded as equal to 0.

Data other than 0 and <empty> is regarded as equal to 1.

Data less than 0.00000001, however, is regarded as undefined.



5. Beam radius compensation

Offset No.	Variable No.
1	#2000
2	#2001
⋮	⋮
64	#2063

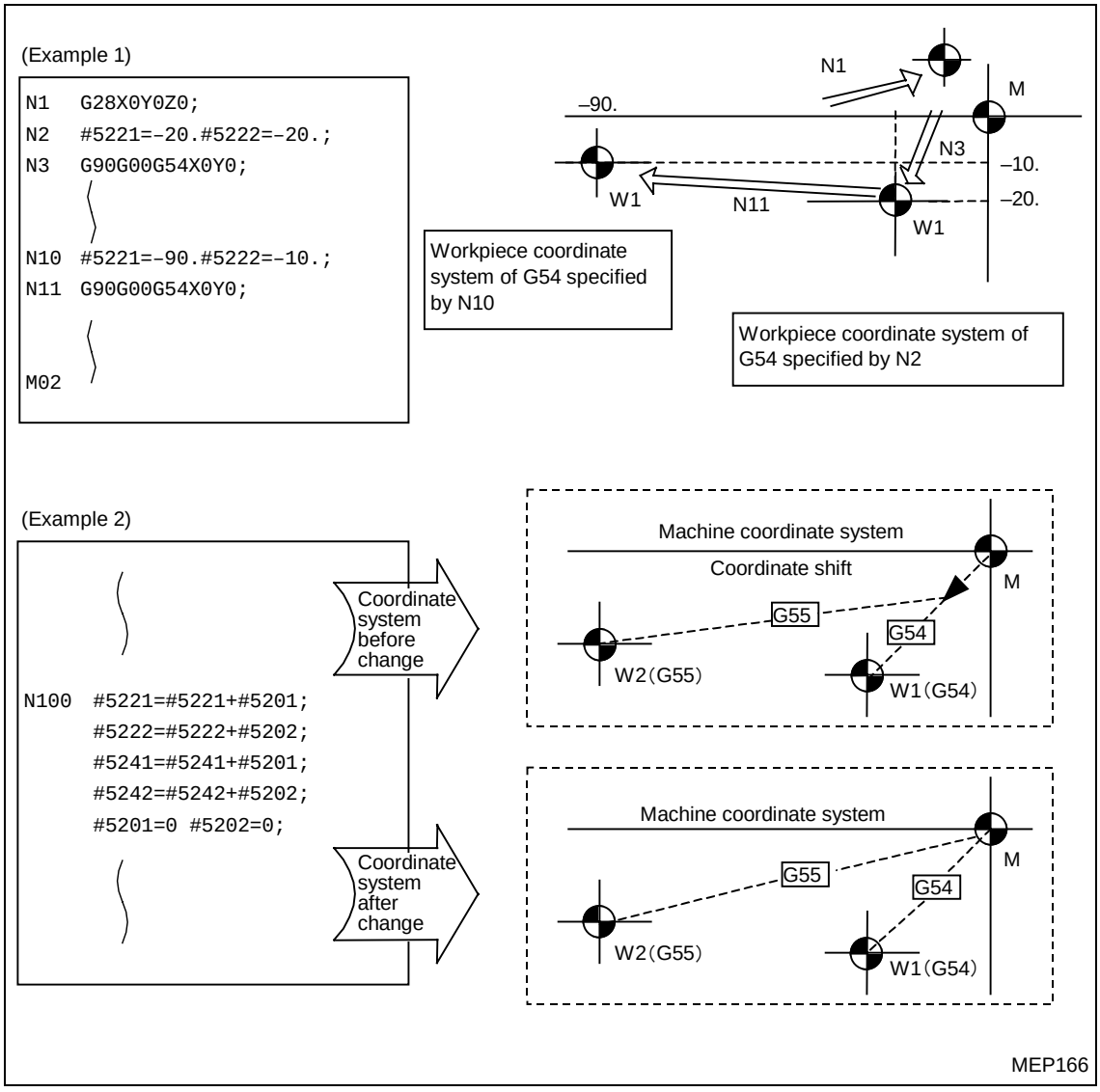
Using variables numbers, you can read beam radius compensation data or assign data.
As with other variables, beam radius compensation data is to contain the decimal point. The decimal point must therefore be included if you want to set data that has decimal digits.

6. Workpiece coordinate system offset

Using variables numbers from 5201 to 5323, you can read workpiece coordinate system offset data or assign data.

Note: The number of controllable axes depends on the NC specifications.

Axis No. / Coordinate name	1st axis	2nd axis	3rd axis
SHIFT	#5201	#5202	#5203
G54	#5221	#5222	#5223
G55	#5241	#5242	#5243
G56	#5261	#5262	#5263
G57	#5281	#5282	#5283
G58	#5301	#5302	#5303
G59	#5321	#5322	#5323



The example 2 shown above applies only when coordinate shift data is to be added to the offset data of a workpiece coordinate system (G54 or G55) without changing the position of the workpiece coordinate system.

7. NC alarm (#3000)

The NC unit can be forced into an alarm status using variables number 3000.

```
#3000 = 70 ;
        └──────── Alarm No.
```

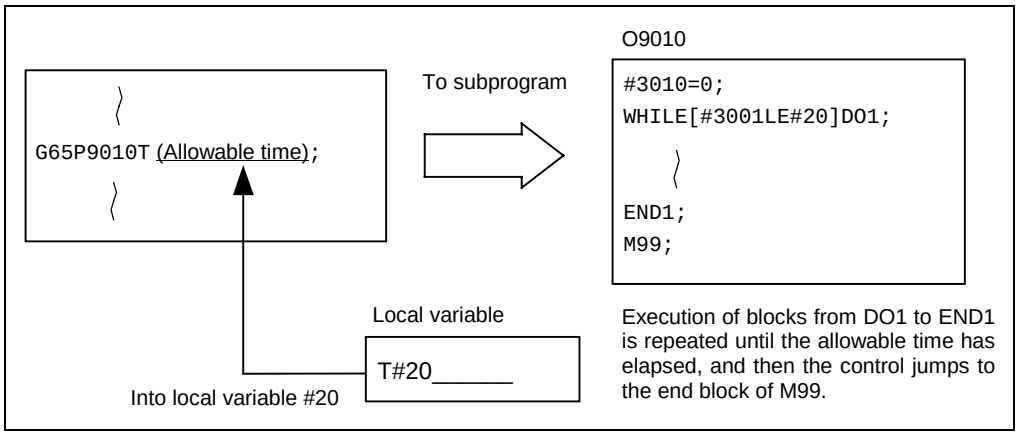
The setting range for the alarm No. is from 1 to 100.

8. Integrated time (#3010, #3015)

Using variables #3010 and #3015, you can read the integrated time existing during automatic operation or assign data.

Type	Variable No.	Unit	Data at power-on	Initialization	Counting
Integrated time 1	3010	sec	Same as at power-off	Data is assigned in variables.	Always during power-on
Integrated time 2	3015				During auto-operation

The integrated time is cleared to 0 after having reached about 2.44×10^{11} msec (about 7.7 years).



9. Suppression of single-block stop (#3003)

Assigning one of the values listed in the table below to variables number 3003 allows single-block stop to be suppressed at subsequent blocks.

#3003	Single block stop
0	No suppression
1	Suppression

10. Validation/invalidation of feed hold, feed rate override, or G09 (#3004)

Feed hold, feed rate override, or G09 can be made valid or invalid for subsequent blocks by assigning one of the values listed in the table below to variables number 3004.

#3004 / Contents (Value)	Bit 0 Feed hold	Bit 1 Feed rate override	Bit 2 G09 check
0	Effective	Effective	Effective
1	Ineffective	Effective	Effective
2	Effective	Ineffective	Effective
3	Ineffective	Ineffective	Effective
4	Effective	Effective	Ineffective
5	Ineffective	Effective	Ineffective
6	Effective	Ineffective	Ineffective
7	Ineffective	Ineffective	Ineffective

Note: Each of the listed bits makes the function valid if 0, or invalid if 1.

11. Parts count

The variable numbered 3005 can be used to read or assign the total number of machined parts. The number in question is internally counted up in normal operation by the execution of the miscellaneous function M02 or M30. Describe a macro instruction as follows to forcibly count up the number by one in the flow of the program.

```
#3005=#3005+1;
```

12. Mirror image (#3007)

The mirror image status of each axis at one particular time can be checked by reading variables number 3007.
Variable #3007 has its each bit crosskeyed to an axis, and these bits indicate that:
If 0, the mirror image is invalid.
If 1, the mirror image is valid.

#3007

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Axis no.													4	3	2	1

13. G-function modal status

The modal status of the block being executed can be checked by reading variables numbered from #4201 to #4220.

Variable Nos. Block executed	Function	Variable Nos. Block executed	Function
#4201	G-code (group 01)	#4211	G-code (group 11)
#4202	G-code (group 02)	#4212	G-code (group 12)
#4203	G-code (group 03)	#4213	G-code (group 13)
#4204	G-code (group 04)	#4214	G-code (group 14)
#4205	G-code (group 05)	#4215	G-code (group 15)
#4206	G-code (group 06)	#4216	G-code (group 16)
#4207	G-code (group 07)		G-code (group 17)
#4208	G-code (group 08)		G-code (group 18)
#4209	G-code (group 09)	#4219	G-code (group 19)
#4210	G-code (group 10)	#4220	G-code (group 20)

14. Other modal information

The modal information of the block being executed can be checked by reading variables numbered from #4301 to #4320.

Variable Nos. / Block executed	Modal information	Variable Nos. / Block executed	Modal information
#4301		#4311	
#4302		#4312	
#4303		#4313	
#4304		#4314	Sequence No.…N
#4305		#4315	Program No.…O
#4306		#4316	
#4307	Beam radius comp. No.…D	#4317	
#4308		#4318	
#4309	Rate of feed …F	#4319	
#4310		#4320	

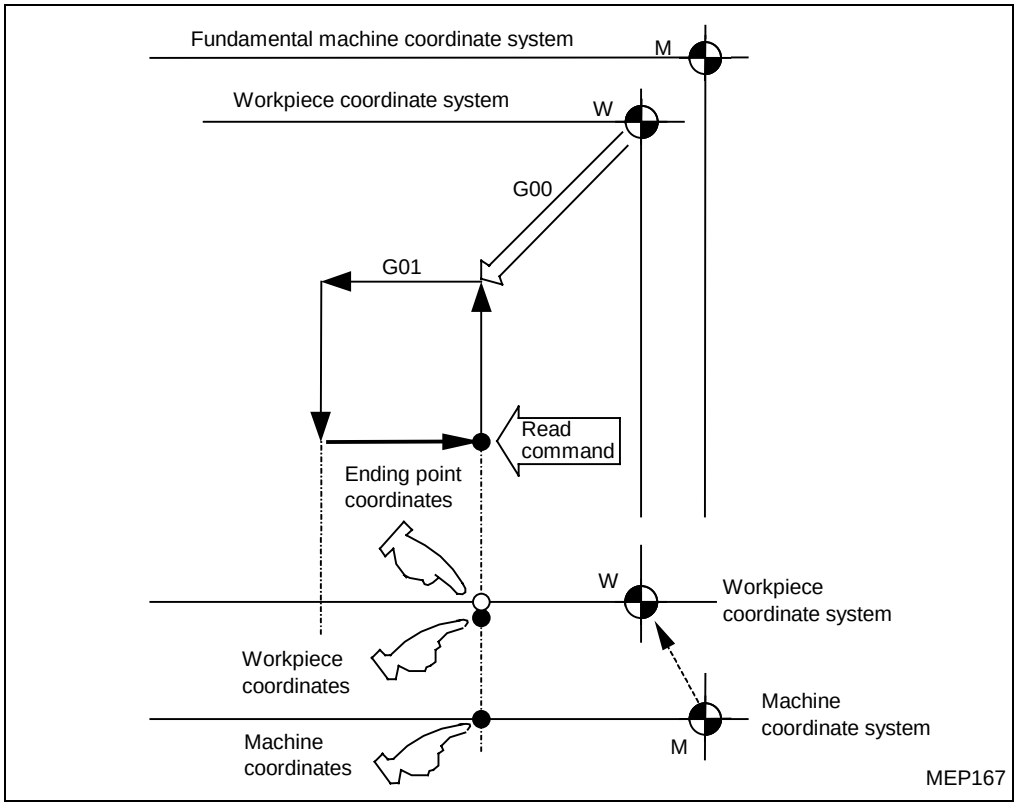
15. Position information

Using variables numbers from #5001 to #5104, you can check the ending-point coordinates of the previous block, machine coordinates, workpiece coordinates, skip coordinates, and servo deviations.

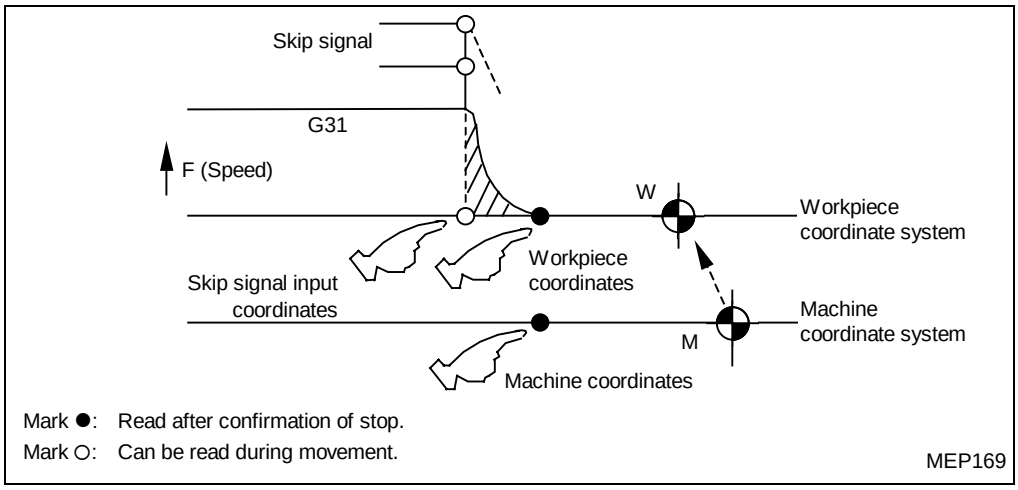
Position information / Axis. No.	Ending point coordinates of previous block	Machine coordinate	Workpiece coordinate	Skip coordinate	Servo deviation amount
1	#5001	#5021	#5041	#5061	#5101
2	#5002	#5022	#5042	#5062	#5102
3	#5003	#5023	#5043	#5063	#5103
4	#5004	#5024	#5044	#5064	#5104
Remarks (Reading during move)	Possible	Impossible	Impossible	Possible	Impossible

Note: The number of controllable axes depends on the machine specifications.

1. The ending-point coordinates and skip coordinates are values in the workpiece coordinate system.
2. Ending-point coordinates, and skip coordinates can be checked even during movement. Machine coordinates, workpiece coordinates, and servo deviations must be checked only after movement has stopped.

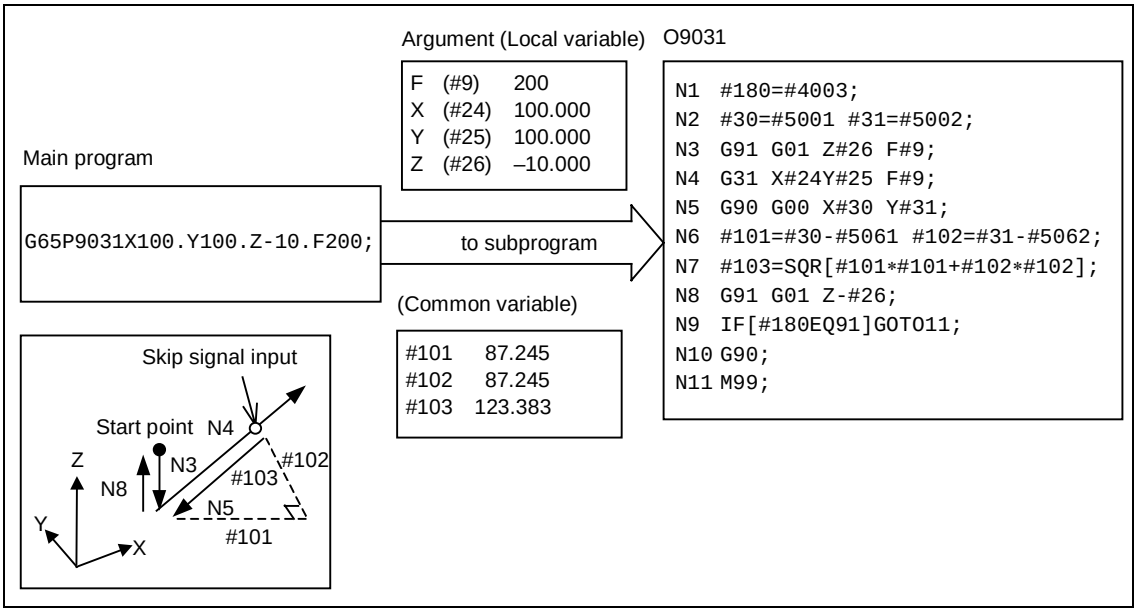


3. Skip coordinates denote the position at which the skip signal has turned on during execution of a G31 block. If the skip signal has not turned on, skip coordinates will denote the corresponding ending-point.
4. The ending-point position denotes the tool tip position which does not allow for any tool offsets, whereas machine coordinates, workpiece coordinates, and skip coordinates denote the tool reference-point position which allows for tool offsets.



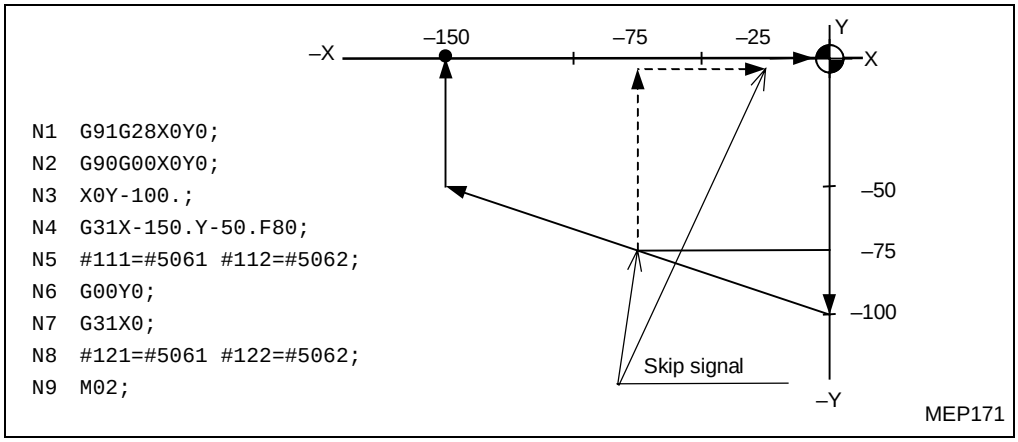
The skip-signal input coordinates are values in the workpiece coordinate system. The coordinates stored in variables from #5061 to #5063 are those existing when the skip signal was input during movement. These coordinates can therefore be read at any time after that.

Example 1: Workpiece position measurement:
The following shows an example of measuring the distance from a measurement reference point to the workpiece end:



#101	X-axis measuring distance	N1	Storage of modal condition about G90/G91
#102	Y-axis measuring distance	N2	Storage of X- and Y-axis starting point coordinates
#103	Linear measuring distance	N3	Infeed on the Z-axis
#5001	Measurement starting point's X-coordinate	N4	Measurement in the XY-plane (Stop at skip signal input)
#5002	Measurement starting point's Y-coordinate	N5	Return to the starting point in the XY-plane
#5061	Skip signal input point's X-coordinate	N6	Calculation of X- and Y-axis measuring distances
#5062	Skip signal input point's Y-coordinate	N7	Calculation of linear measuring distance
		N8	Escape on the Z-axis
		N9, N10	Resumption of modal condition about G90/G91
		N11	Return from subprogram

Example 2: Reading skip signal input coordinates:



#111 = –75. + ε #112 = –75. + ε
#121 = –25. + ε #122 = –75. + ε

where ε denotes an error due to response delay.
Variable #122 denotes the skip signal input coordinate of N4 since N7 does not have any Y-axis command.

6-3-3 Arithmetic operation commands

Various operations can be carried out between variables using the following format.

#i = <expression>

where <expression> must consist of a constant(s), a variable(s), a function(s), or an operator(s). In the table given below, constants can be used instead of #j and/or #k.

[1] Definition/replacement of variables	`#i=#j`	Definition/replacement
[2] Additional-type operations	`#i=#j+#k`	Addition
	`#i=#j-#k`	Subtraction
	`#i=#jOR#k`	Logical additon (For each of 32 bits)
	`#i=#jXOR#k`	Exclusive OR (For each of 32 bits)
[3] Multiplicative-type operations	`#i=#j*#k`	Multiplication
	`#i=#j/#k`	Division
	`#i=#jMOD#k`	Surplus
	`#i=#jAND#k`	Logical product (For each of 32 bits)
[4] Functions	`#i=SIN[#k]`	Sine
	`#i=COS[#k]`	Cosine
	`#i=TAN[#k]`	Tangent (tanθ is used as sinθ/cosθ.)
	`#i=ATAN[#j]`	Arc-tangent (Either ATAN or ATN can be used.)
	`#i=ACOS[#j]`	Arc-cosine
	`#i=SQRT[#k]`	Square root (Either SQRT or SQR is available.)
	`#i=ABS[#k]`	Absolute value
	`#i=BIN[#k]`	BINARY conversion from BCD
	`#i=BCD[#k]`	BCD conversion from BINARY
	`#i=ROUND[#k]`	Rounding to the nearest whole number (Either ROUND or RND is available.)
	`#i=FIX[#k]`	Cutting away any decimal digits
	`#i=FUP[#k]`	Counting any decimal digits as 1s
	`#i=LN[#k]`	Natural logarithm
	`#i=EXP[#k]`	Exponent with the base of e (= 2.718 ...)

Note 1: In principle, data without a decimal point is handled as data that has a decimal point. (Example: 1 = 1.000)

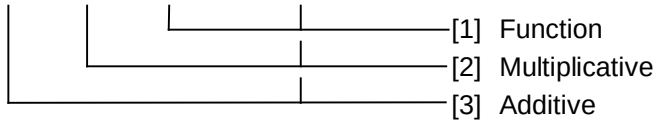
Note 2: Workpiece coordinate system offsets from variable #5201, and other data are data that have a decimal point. If data without a decimal point is defined using these variables numbers, therefore, a decimal point will also be assigned to the data.

Note 3: The <expression> after a function must be enclosed in brackets ([ ]).

1. Operation priority

Higher priority is given to functions, multiplicative operations, and additive operations, in that order.

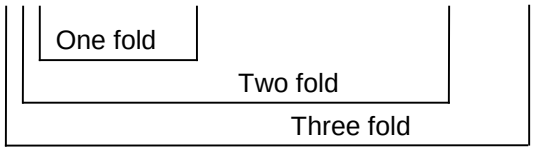
```
#101=#111+#112*SIN[#113]
```



2. Specifying an operational priority level

The part to which the first level of operation priority is to be given can be enclosed in brackets ([ ]).

```
#101=SQRT[[[#111–#112]*SIN[#113]+#114]*#15];
```



3. Examples of operation instructions

[1] Main program and argument specification	`G65 P100 A10 B20.`
	
	`#101=100.000`
	`#102=200.000`
		#1 10.000
		#2 20.000
		#101 100.000
		#102 200.000

[2] Definition, replacement =	`#1=1000`
	`#2=1000.`
	`#3=#101`
	`#4=#102`
	`#5=#5081`
		#1 1000.000
		#2 1000.000
		#3 100.000 } Data of common variables
		#4 200.000 }
		#5 –10.000 Offset amount

[3] Addition, subtraction + –	`#11=#1+1000`
	`#12=#2–50.`
	`#13=#101+#1`
	`#14=#5081–3.`
	`#15=#5081+#102`
		#11 2000.000
		#12 950.000
		#13 1100.000
		#14 –13.000
		#15 190.000

[4] Logical addition OR	`#3=100` `#4=#3OR14`	#3 = 01100100 14 = 00001110 #4 = 01101110 = 110
[5] Exclusive OR XOR	`#3=100` `#4=#3XOR14`	#3 = 01100100 14 = 00001110 #4 = 01101010 = 106

[6] Multiplication, Division * /	`#21=100*100`
	`#22=100.*100`
	`#23=100*100.`
	`#24=100.*100.`
	`#25=100/100`
	`#26=100./100`
	`#27=100/100.`
	`#28=100./100.`
	`#29=#5081*#101`
	`#30=#5081/#102`
		#21 10000.000
		#22 10000.000
		#23 10000.000
		#24 10000.000
		#25 1.000
		#26 1.000
		#27 1.000
		#28 1.000
		#29 –1000.000
		#30 –0.050

[7] Surplus MOD	`#31=#19MOD#20`	#19 / #20 = 48 / 9 = 5 surplus 3
[8] Logical product AND	`#9=100` `#10=#9AND15`	# 9 = 01100100 15 = 00001111 #10 = 00000100 = 4

[9] Sine SIN	`#501=SIN[60]`
	`#502=SIN[60.]`
	`#503=1000*SIN[60]`
	`#504=1000*SIN[60.]`
	`#505=1000.*SIN[60]`
	`#506=1000.*SIN[60.]`
	Note: SIN[60] is equal to SIN[60.].
		#501 0.866
		#502 0.866
		#503 866.025
		#504 866.025
		#505 866.025
		#506 866.025

[10] Cosine COS	`#541=COS[45]`
	`#542=COS[45.]`
	`#543=1000*COS[45]`
	`#544=1000*COS[45.]`
	`#545=1000.*COS[45]`
	`#546=1000.*COS[45.]`
	Note: COS[45] is equal to COS[45.].
		#541 0.707
		#542 0.707
		#543 707.107
		#544 707.107
		#545 707.107
		#546 707.107

[11] Tangent TAN	#551=TAN[60] #552=TAN[60.] #553=1000*TAN[60] #554=1000*TAN[60.] #555=1000.*TAN[60] #556=1000.*TAN[60.] **Note:** TAN[60] is equal to TAN[60.].	#551 1.732 #552 1.732 #553 1732.051 #554 1732.051 #555 1732.051 #556 1732.051
[12] Arc-tangent ATAN	#561=ATAN[173205/100000] #562=ATAN[173.205/100.] #563=ATAN[1.732]	#561 60.000 #562 60.000 #563 59.999
[13] Arc-cosine ACOS	#521=ACOS[100000/141421] #522=ACOS[100./141.421] #523=ACOS[1000/1414.213] #524=ACOS[10./14.142] #525=ACOS[0.707]	#521 45.000 #522 45.000 #523 45.000 #524 44.999 #525 45.009
[14] Square root SQRT	#571=SQRT[1000] #572=SQRT[1000.] #573=SQRT[10.*10.+20.*20.] #574=SQRT[#14*#14+#15*#15]	#571 31.623 #572 31.623 #573 22.361 #574 190.444
[15] Absolute value ABS	#576=–1000 #577=ABS[#576] #3=70. #4=–50. #580= ABS[#4–#3]	#576 –1000.000 #577 1000.000 #580 120.000
[16] BIN, BCD	#1=100 #11=BIN[#1] #12=BCD[#1]	 #11 64 #12 256
[17] Rounding into the nearest whole number ROUND	#21=ROUND[14/3] #22=ROUND[14./3] #23=ROUND[14/3.] #24=ROUND[14./3.] #25=ROUND[–14/3] #26=ROUND[–14./3] #27=ROUND[–14/3.] #28=ROUND[–14./3.]	#21 5 #22 5 #23 5 #24 5 #25 –5 #26 –5 #27 –5 #28 –5
[18] Cutting away any decimal digits FIX	#21=FIX[14/3] #22=FIX[14./3] #23=FIX[14/3.] #24=FIX[14./3.] #25=FIX[–14/3] #26=FIX[–14./3] #27=FIX[–14/3.] #28=FIX[–14./3.]	#21 4.000 #22 4.000 #23 4.000 #24 4.000 #25 –4.000 #26 –4.000 #27 –4.000 #28 –4.000
[19] Counting any decimal digits as 1s FUP	#21=FUP[14/3] #22=FUP[14./3] #23=FUP[14/3.] #24=FUP[14./3.] #25=FUP[–14/3] #26=FUP[–14./3] #27=FUP[–14/3.] #28=FUP[–14./3.]	#21 5.000 #22 5.000 #23 5.000 #24 5.000 #25 –5.000 #26 –5.000 #27 –5.000 #28 –5.000
[20] Natural logarithm LN	#101=LN[5] #102=LN[0.5] #103=LN[–5]	#101 1.609 #102 –0.693 Error 859 CALCULATE IMPOSSIBLE
[21] Exponent EXP	#104=EXP[2] #105=EXP[1] #106=EXP[–2]	#104 7.389 #105 2.718 #106 0.135

4. Operation accuracy

The errors listed in the table below are generated by one arithmetic operation, and the error rate increases each time an operation is performed.

Operation format	Mean error	Max. error	Kind of error
a = b + c a = b – c	2.33×10^{-10}	5.32×10^{-10}	Min. $\left\lvert \frac{\varepsilon}{b} \right\rvert$, $\left\lvert \frac{\varepsilon}{c} \right\rvert$
a = b • c	1.55×10^{-10}	4.66×10^{-10}	Relative error $\left\lvert \frac{\varepsilon}{a} \right\rvert$
a = b/c	4.66×10^{-10}	1.86×10^{-9}	
a = $\sqrt{b}$	1.24×10^{-9}	3.73×10^{-9}	
a = sin b a = cos b	5.0×10^{-9}	1.0×10^{-8}	Absolute error $\lvert \varepsilon \rvert$ degree
a = $\tan^{-1}$ b/c	1.8×10^{-6}	3.6×10^{-6}	

Note: The function TAN (Tangent) is calculated as SIN/COS (Sine/Cosine).

5. Notes on deterioration of accuracy

A. Addition/subtraction

As for additional-type operations, if an absolute value is subtracted from the other, the relative error cannot be reduced below 10^{-8}.

For example, when the true values (such values, by the way, cannot be substitued directly) of #10 and #20 are as follows:

#10 = 2345678988888.888
#20 = 2345678901234.567

then #10 – #20 = 87654.321 would not result from calculation of #10 – #20. This is because, since the effective number of digits of the variable is eight (decimal), the approximate values of #10 and #20 are:

#10 = 2345679000000.000
#20 = 2345678900000.000

More strictly, internal binary values slightly differ from these values. Actually therefore, a significant error results as follows:

#10 – #20 = 100000.000.

B. Logical relationship

As for EQ, NE, GT, LT, GE and LE, the processing is executed in a similar manner to addition and substraction, so be careful to errors. For example, to judge whether #10 is equal to #20 of the above example, the conditional expression

IF [#10EQ#20]

is not appropriate due to the errors. In such a case, therefore, give a macro-instruction as shown below to allow for an acceptable tolerance in the judgement on the equality of two values.

IF [ABS[#10 – #20] LT200000]

C. Trigonometric functions

For trigonometric functions, although the absolute error is guaranteed, the relative error is not below 10^{-8}. Be careful, therefore, when carrying out multiplication, or division after trigonometric function operations.

6-4 Control Commands

The flow of a program can be controlled using IF ~ GOTO ~ and WHILE ~ DO ~ commands.

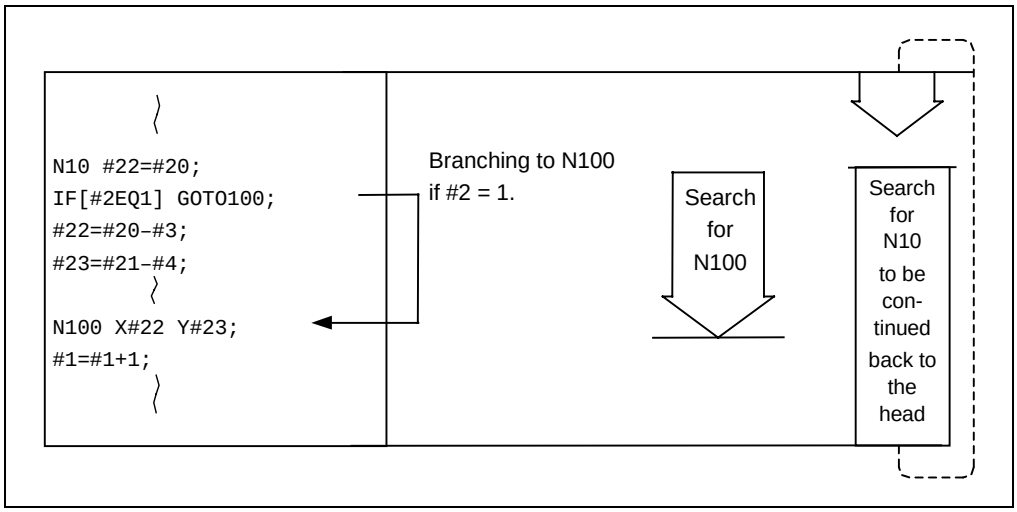
1. Branching

Format: IF [conditional expression] GOTO n

where n is a sequence number in the same program.
The branching will occur to the block headed by sequence number 'n' if the condition holds, or if the condition does not hold, the next block will be executed.
An independent setting of GOTO statement without IF [conditional expression] will perform unconditional branching to the specified block.
The [conditional expression] consists of the following six types:

#i EQ #j	=	(#i is equal to #j.)
#i NE #j	≠	(#i is not equal to #j.)
#i GT #j	>	(#i is larger than #j.)
#i LT #j	<	(#i is smaller than #j.)
#i GE #j	≥	(#i is equal to #j, or larger than #j.)
#i LE #j	≤	(#i is equal to #j, or smaller than #j.)

For GOTO n, n must be a sequence number within the same program. If the sequence number does not exist in that program, an alarm will occur. An expression or a variable can be used instead of #i, #j, or n.
Sequence number designation Nn must be set at the beginning of the destination block. Otherwise, an alarm will result. If, however, the block begins with "/" and Nn follows, the program can be branched into that sequence number.



Note: During search for the branching destination sequence number, if the area from the block after "IF ..." to the program end (% code) is searched (forward search) in vain, then the area from the head down to the block before "IF ..." will be searched next (backward search). It will therefore take more time to execute backward search (searching in the opposite direction to the flow of the program) than to execute forward search.

2. Looping

Format:

```
WHILE [Condition expression] DOm;   (m = 1, 2, 3 ··· 9999)
    ≀
ENDm
```

The area from the next block to the ENDm block loops while the conditional expression holds. If the conditional expression does not hold, control will be tansferred to the block after ENDm. In the format shown above, DOm can precede WHILE.

You must always use WHILE [conditional expression] DOm and ENDm in pairs. If you omit WHILE [conditional expression], the area from DOm to ENDm will endlessly loop. In DOm, m (1 to 9999) identifies the number of looping. (DO1, DO2, DO3, and so on up to DO9999)

The maximum available number of degrees of multiplicity is 27.

[1] Same identifying No. can be used repeatedly.

Usable

```
WHILE ~ DO1
END1
```

Usable

```
WHILE ~ DO1
END1
```

[2] The identifying No. of WHILE ~ DOm is arbitrary.

Usable

```
WHILE ~ DO1
    ≀
END1
    ≀
WHILE ~ DO3
    ≀
END3
    ≀
WHILE ~ DO2
    ≀
END2
    ≀
WHILE ~ DO1
    ≀
END1
```

[3] WHILE ~ DOm must not overlap.

Unusable

```
WHILE ~ DO1
    ≀
WHILE ~ DO2
    ≀
END1
    ≀
END2
```

[4] Branching into WHILE ~ DOm is not allowed.

Unusable

```
WHILE ~ DO1
N2·····
END1
IF ~ GOTO2
```

[5] Outward branching from the range of WHILE ~ DOm is possible.

Usable

```
WHILE ~ DO1
IF ~ GOTO2
END1
N2·····
```

7 PROGRAMMABLE FOCUS POSITIONING (SERVO FOCUS)

7-1 Overview

Owing to a controlled axis (named W) added for changing the position of the lens in the torch, the function of programmable focus positioning (Servo Focus) allows the position of the focal point to be changed easily in the flow of program execution.
This chapter mainly explains how to give W-axis motion commands.

Note: This function is an improvement introduced in the CNC, version v60.2.

7-2 Interpolation Functions

7-2-1 Positioning (Rapid traverse): G00

1. Function and purpose

Positioning command G00 performs a rapid positioning to the ending point specified by a coordinate word.

2. Programming format

```
G00  Xx Yy Zz Ww;
```

where x, y, z and w each denote a coordinate. The absolute or the incremental data input is used according to the status of G90/G91 existing at the particular time.

3. Detailed description

1. Motion commands under G00 are generally executed with respect to the workpiece coordinate system. The workpiece coordinates of the W-axis depend upon the current focal length. Refer to the relevant manual of the INTELLIGENT LOOP SYSTEM for details on the setting of the focal length.
2. Once this command has been given, the G00 mode will be retained until any other G-code command that overrides this mode, that is, G01, G02, or G03 of group 01 is given. Thus, it is merely required to input coordinate words for G00 movements in the succeeding blocks.
3. In the G00 mode, acceleration/deceleration always takes place at the starting/ending point of a block and, after an in-position status check, the program proceeds to the next block.

4. Sample programs

```
G90  G00 W-2.0;
```

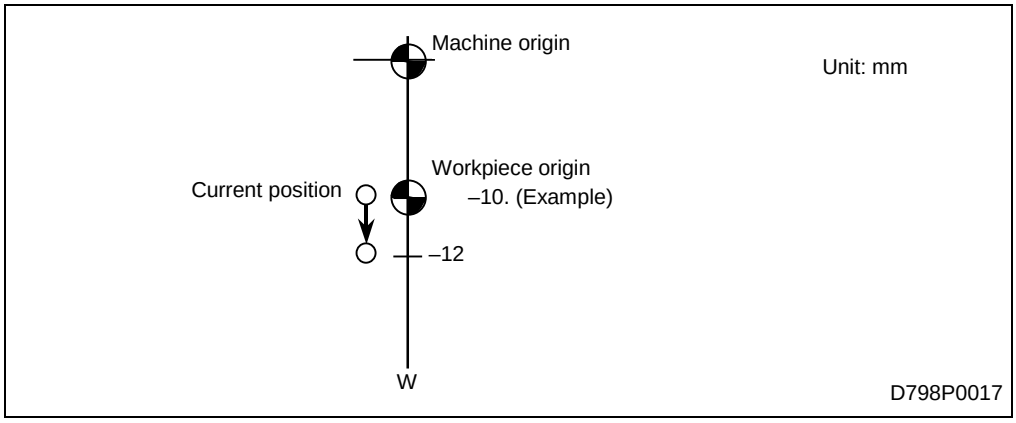


Fig. 7-1

Note 1: Refer to the relevant manual of the INTELLIGENT LOOP SYSTEM for details on the setting of the focal length.

Note 2: Refer to the specifications of the particular machine for the actual rate of rapid traverse.

Note 3: The “tool” will be moved by a G00 command at the rapid traverse rate.
If a rapid traverse rates of 24000 and 3000 mm/min are preset respectively for the X- and W-axes, then the command

```
G91  G00 X–400. W100.;
```

will be executed as illustrated below.

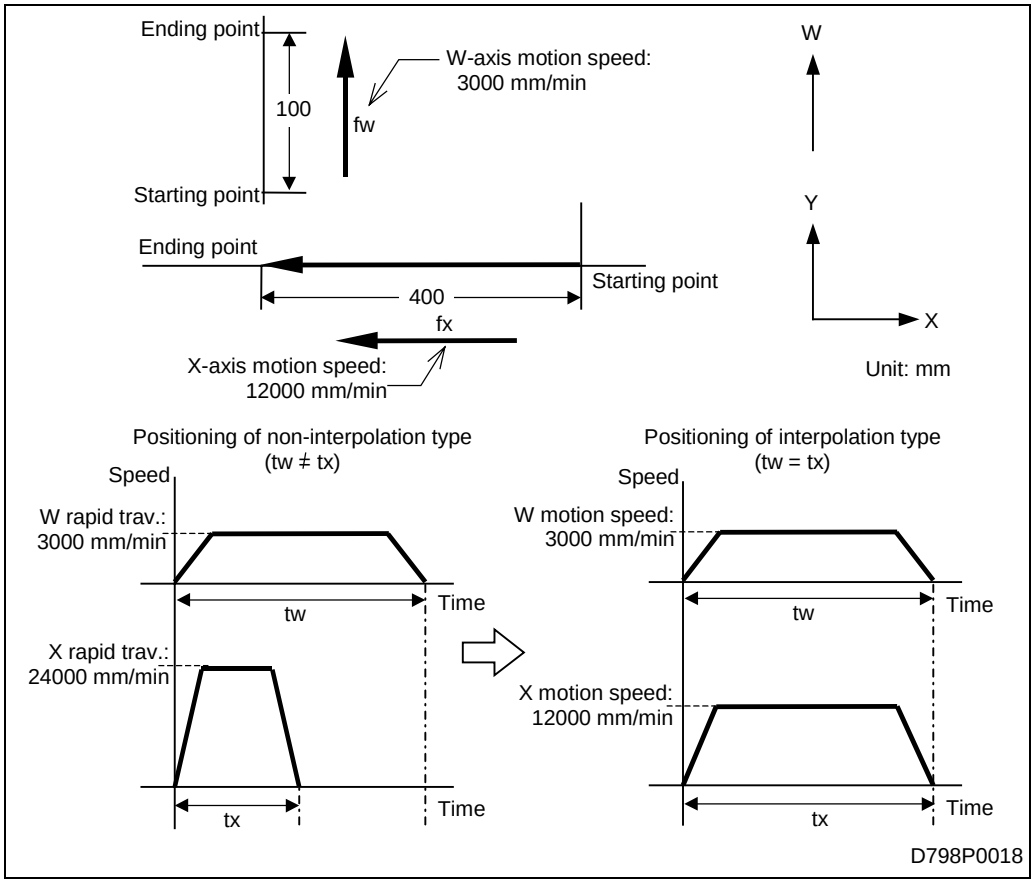


Fig. 7-2

7-2-2 Linear interpolation: G01

1. Function and purpose

This command moves (interpolates) a tool from the current position to the ending point specified by a coordinate word, at the rate of feed specified with address F. The specified speed acts here as the linear velocity relative to the direction of movement of the tool center.

2. Programming format

```
G01  Xx Yy Zz Ww Ff;
```

where x, y, z and w each denote a coordinate. The absolute or the incremental data input is used according to the status of G90/G91 existing at the particular time.

3. Detailed description

1. Motion commands under G01 are generally executed with respect to the workpiece coordinate system. The workpiece coordinates of the W-axis depend upon the current focal length. Refer to the relevant manual of the INTELLIGENT LOOP SYSTEM for details on the setting of the focal length.
2. Once this command has been given, the G01 mode will be retained until any other G-code command that overrides this mode, that is, either G00, G02, or G03 of command group 01 is given. Thus, it is merely required to input coordinate words for linear interpolations in the succeeding blocks unless the rate of feed must be changed.
   A programming error will result if no F-code command has been given to the first G01 command.

7-2-3 Reference point return: G28

1. Function and purpose

G28 command first performs a G00 (rapid) positioning to the specified intermediate point along the specified axes, and then a returning to the first reference point at the rapid traverse rate independently along each specified axis.

2. Programming format

```
G28  Xx Yy Zz Ww;
```

Use the addresses X, Y, Z and W to specify the coordinates of the intermediate point as required.

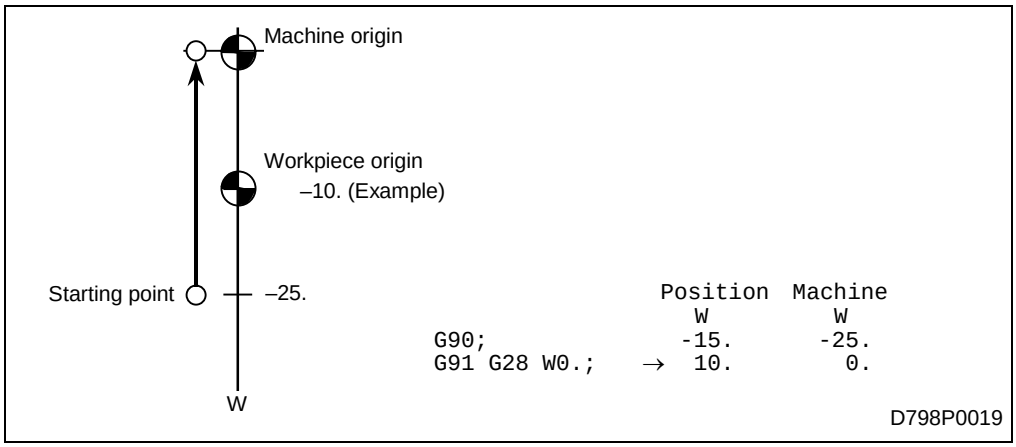


Fig. 7-3

7-2-4 Machine coordinate system selection: G53

1. Function and purpose

Use the G53 command, which is executed independently of the currently active workpiece coordinate system, to perform a positioning motion to a point which is easier to be described in the machine coordinate system.

2. Programming format

```
G53  Xx Yy Zz Ww;
```

3. Detailed description

The function G53, which is non-modal, is provided for a rapid positioning to the target point that is to be specified in machine coordinates (absolute data input [G90]) in the same block with G53.

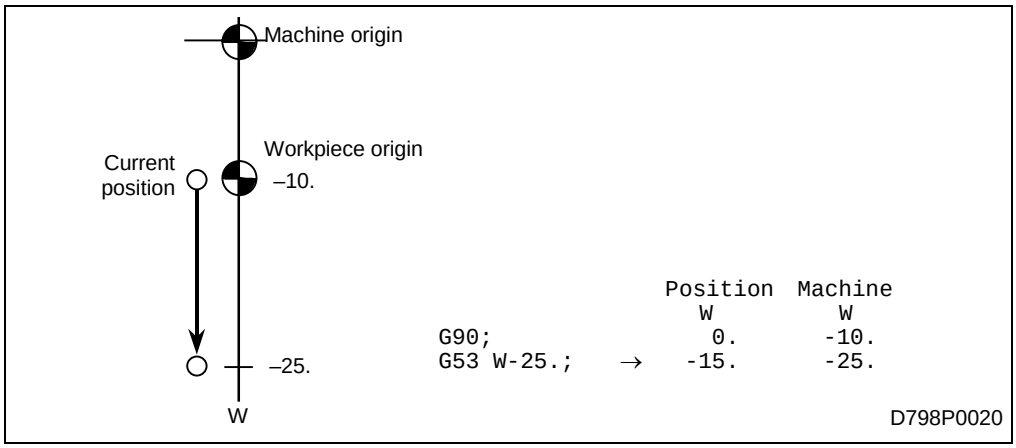


Fig. 7-4